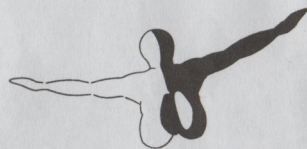
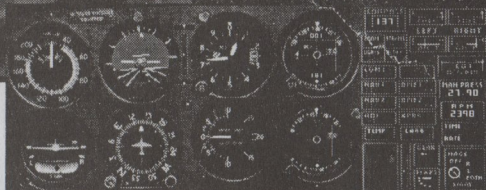
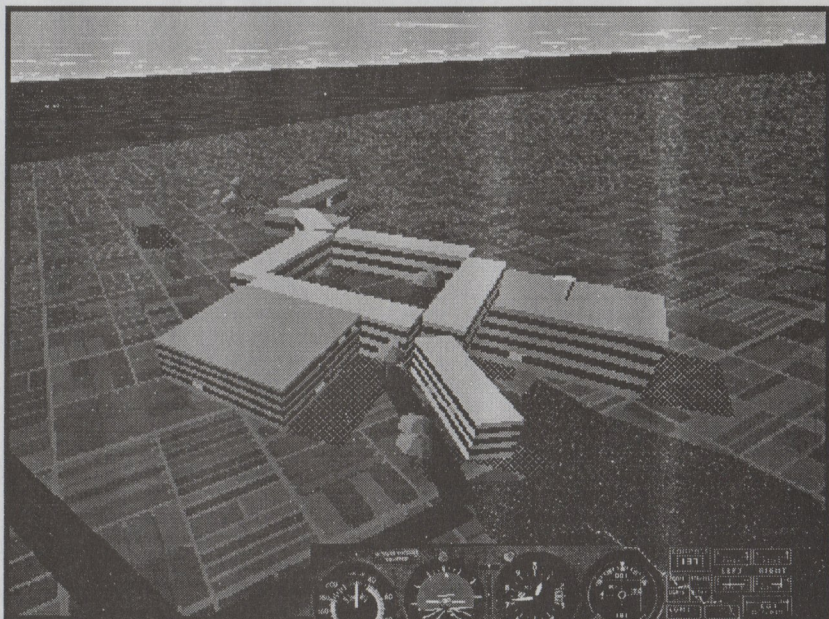


FS5 Designer-Tool



aerosoft



für Microsoft Flight Simulator FS5

FS5 Designer Tool 1.1

Werkzeug zum Erzeugen von Szenieriedateien
für Microsoft® Flight Simulator® 5.0

Handbuch

FS5 Designer Tool, Version 1.1

© 1995

Aerosoft GmbH Luftfahrt-Datentechnik,
Autor: Andreas de Hair
Flughafen Paderborn/Lippstadt,
D-33142 Büren-Ahden

Für die Mitteilung von Fehlern ist der Autor dankbar.

Microsoft, MS und MS-DOS sind eingetragene Warenzeichen der Microsoft Corporation.

Flight Simulator ist ein eingetragenes Warenzeichen von Bruce Artwick.

FS5 Designer Tool 1.1 wurde mit Borland C++ 3.0 entwickelt.

Mit dem FS5 Designer Tool erstellte Szenieriedateien dürfen ohne Einwilligung der Microsoft Corporation und der Aerosoft^{GmbH}, Büren nur für den privaten Gebrauch genutzt werden! Vermarktung und Verkauf sind untersagt.

Inhalt

1	Vorwort.....	4
2	Erste Schritte.....	5
2.1	Voraussetzungen und Installation.....	5
2.2	Das Grundgerüst jedes FS5DesignerTool-Programms	6
2.3	Koordinatendarstellungen.....	6
2.3.1	Absolute Koordinate.....	6
2.3.2	Relative Koordinate.....	7
2.3.3	Blockkoordinate.....	7
2.4	Numerische Ausdrücke	7
2.5	Strings	8
2.6	Kommentare	8
3	Flugnavigation	9
3.1	VOR.....	9
3.2	ILS	10
3.3	NDB.....	10
3.4	IMO Marker.....	11
4	Einfache Grafikoperationen.....	12
4.1	Sichtbarkeitsbereiche.....	12
4.2	Farben	13
4.3	Linien und Straßen	13
4.4	Flächen - Mal gefüllt, mal farbig	16
4.5	Bedingungen und Unterprogramme	20
4.6	Weitere einfache Grafikoperationen	24
5	Landschaftsbau einmal einfach - Die Sectionen 1 bis 6.....	27
5.1	Die sechs Sectionen.....	27
5.2	Landschaftstypen	27
5.3	Das Streifen-Konzept	28
5.4	Brauchen wir Grenzen?.....	28
5.5	Gebirgetypen	29
6	Flughäfen und Landebahnen.....	30
6.1	AIRPORT und GEO_AREA.....	30
6.2	RUNWAY.....	30
6.3	ATIS.....	32
7	Ablaufstrukturen	34
7.1	Der.....	34
7.2	Der.....	35
8	Die FS5 Szeneriebibliothek - Bedeutung der IDs	36
9	Speicherprobleme	37
ANHANG		38
A	Alphabetische Liste der FS5DesignerTool-Befehle.....	38
B	Beispielprogramme.....	50
C	Umrechnung absoluter Koordinaten in Blockkoordinaten	51
D	Index.....	52

1. Vorwort

Lieber Leser, liebe Leserin,

mit dem FS5DesignerTool liegt Ihnen das Ergebnis einer nun einjährigen Entwicklungsarbeit vor. Ziel der Entwicklung war es, eine möglichst einfache Erstellung von FS5-kompatiblen Szenarien zu ermöglichen. Grundlage unserer Arbeit waren die Szeneriespezifikationen, die der Microsoft Flight Simulator der Versionen 5.0 und 5.0a verarbeiten konnte.

In der vorliegenden Version haben wir bereits einige Ihrer Verbesserungsvorschläge realisieren können. Und, das neue Programm arbeitet wesentlich schneller als die "alte" Version 1.0. Neue Befehle wurden implementiert, und schließlich konnte auch das Handbuch nachgebessert werden. Wir hoffen, daß Ihnen die Arbeit mit dem FS5DesignerTool noch leichter fällt.

Selbst wenn abzusehen ist, daß in bald eine neue Version des Flight Simulator erscheinen wird, ist der Autor der Meinung, daß Szeneriedesign für den FS5.0 die Flight Simulator-Gemeinde noch eine ganze Weile in Atem halten wird. Schließlich liegen die Fähigkeiten des FS5.0 weit über jenen, die der FS4 bot, und Szeneriedesign für den FS5 befindet sich immer noch in den Kinderschuhen.

Wie lange das so bleibt, hängt auch von Ihnen, liebe Leser, und Ihren eigenen Aktivitäten im FS5-Szenarienbau ab. Wir hoffen, daß Sie das FS5DesignerTool genauso schätzen werden, wie wir es bereits tun.

Für ein erfolgreiches Arbeiten mit dem FS5DesignerTool und auch für das Durcharbeiten dieses Handbuchs ist es auf jeden Fall unerläßlich, daß Sie sich bereits mit den einfacheren Begriffen aus der "FS5-Welt" vertraut gemacht haben. So weit, daß Sie also z.B. bereits wissen, was ein NDB ist oder welche Typen von Anflugbefeuerungen es bei Landebahnen gibt (näheres im Handbuch des FS5).

Eine letzte Anmerkung möchte ich mir noch gestatten. Wenn Sie beim Arbeiten mit dem FS5DesignerTool auf Probleme stoßen sollten, oder wenn Sie feststellen, daß bestimmte Befehle des FS5DesignerTools sich als unhandlich im Gebrauch erweisen, so teilen Sie uns das doch bitte kurz mit. Sie erhalten damit die Chance, daß Ihre Ideen direkt in spätere Versionen des FS5DesignerTools integriert werden können.

Zukünftige Updates erhalten Sie über aerosoft, Flughafen Paderborn/Lippstadt, D-33142 Büren.

Und nun viel Spaß beim Szeneriedesign mit dem FS5DesignerTool!

Paderborn, im Januar 1995

Andreas de Hair

Bitte beachten Sie: "Normaler" Text ist immer in Helvetica gesetzt, Programmbeispiele und Direkteingaben unter MS-DOS in Courier.

2. Erste Schritte

2.1 Voraussetzungen und Installation

Was wird zum Einsatz des FS5DesignerTools gebraucht?

Das FS5DesignerTool arbeitet, wie der FS5 selbst, unter MS-DOS, ab Version 5.0 oder später.

Als erstes benötigen Sie ein Programm zum Erstellen der Quelltexte, einen Editor. Der "edit" Befehl des MS-DOS-Betriebssystems sollte in den meisten Fällen ausreichen; ab Version 5.0 gehört er zum Lieferumfang von MS-DOS. Keine Sorge, auch wenn Sie sonst nur eine graphische Benutzeroberfläche kennen: Der "edit" bietet Mausunterstützung und verfügt über eine Online-Hilfe. Dann brauchen Sie natürlich den Microsoft Flight Simulator, in der Version 5.0 bzw. 5.0a. **Ältere Versionen verwenden ein anderes internes Format und können die vom FS5DesignerTool erzeugten Szenieriedaten also nicht lesen.**

Wie wird das FS5DesignerTool installiert?

Um das FS5DesignerTool auf Ihrer Festplatte zu installieren, gehen Sie wie folgt vor.

- 1) Legen Sie die Programmdiskette in Laufwerk A und tippen Sie a:
- 2) Wenn Sie das FS5DesignerTool auf Laufwerk C: installieren wollen, tippen Sie nun ein:
install c: , bei Laufwerk D: geben Sie entsprechend
install d: ein.

Das FS5DesignerTool wird nun im Verzeichnis \fs5dt Ihrer Festplatte installiert.

Zum Starten wechseln Sie mit c: und dann cd \fs5dt in das Arbeitsverzeichnis. Danach geben Sie einmal fst t ein. Wenn das Programm richtig installiert wurde, gibt das FS5DesignerTool während der

"Übersetzung" des Quelltextes drei Dinge aus:

1. den Namen der Datei, die gerade bearbeitet wird
2. die Nummer der Zeile, die gerade übersetzt wird
3. den restlichen freien Speicher, über den das FS5DesignerTool verfügen kann.

Bei "Übersetzen" der ersten Programmzeile sieht die Bildschirmausgabe also etwa so aus:

```
FS5DesignerTool 1.1
t.fst Zeile 1 Mem 490896
```

Diese Ausgabe wird ständig erneuert, so daß Sie gleich einen Eindruck von der Geschwindigkeit des FS5DesignerTools erhalten. Natürlich kann die Größe des freien Speichers von der hier angegebenen Größe abweichen, wenn Sie Ihren Speicher anders konfiguriert haben.

Reicht der Speicher zum Übersetzen des Quelltextes nicht aus, so bricht das FS5DesignerTool mit einer Fehlermeldung ab. Für die meisten Anwendungen dürfte der freie Speicher jedoch genügen. Nur bei sehr großen Szenarien kann es notwendig sein, die Szenarien in mehrere Quelldateien aufzuspalten. Mehr dazu finden Sie im Kapitel "Speicherprobleme".

Das FS5DesignerTool durchläuft nun alle Zeilen des Quelltextes und macht bei erfolgreicher Übersetzung des Quelltextes folgende Ausgabe:

```
FS5DesignerTool 1.1
102 Zeilen gelesen.
VOR/ILS : 4
Section 1 : 1
Primitive Ops: 41
GEO_AREA : 1
NDB : 1
Marker: 3
Schreibe File fsout.bgl
Schreibe File fsout.vis
```

Die beiden so erzeugten Dateien ("fsout.bgl" und "fsout.vis") können Sie sich nun ansehen, indem Sie sie in das scenery-Directory des fs5 kopieren, etwa durch copy fsout.* \fs5\scenery. In diesem Directory stehen alle für den FS5 gültigen Szenarien, die nach Start des FS5 geladen werden können.

Starten Sie, nachdem Sie die beiden Dateien kopiert haben, den FS5. Wählen Sie im Menü "Scenery" den Eintrag "Scenery Library". Dort können Sie dann durch Anklicken von "fsout.vis" die erste eigene Szenerie anwählen.

Wir haben diesen kleinen Umweg über das Verwenden der Bibliotheksfunktion des FS5 gewählt, damit sich die so erzeugte Szenerie nicht mit anderen bestehenden und weitaus umfassenderen Szenerien der FS5-Welt überschneidet.

Nun müssen Sie Ihr Flugzeug nur noch an die definierte Stelle setzen. Am einfachsten erreichen Sie das, indem Sie im "Airport"-Menü den Flughafen "Paderborn-Lippstadt" anwählen.

In dieser Beispielszenerie finden Sie einige der Elemente wieder, die Sie in eigenen Szeneriedateien selber verwenden und variieren können, z.B. Navigationshilfen (VOR, ILS, Marker), einen Flughafen mit komplexer Runway und Gebäuden.

Diese und andere Elemente werden Sie in den nachfolgenden Kapiteln detailliert erläutert finden.

2.2 Das Grundgerüst jedes FS5DesignerTool-Programms

Um mit dem FS5DesignerTool zu starten, müssen Sie nur zwei Befehle kennen.

Jede Szenerie des FS5 erhält eine eigene Kennung, nämlich eine sog. Ident-Nummer. Diese steht in jedem Szeneriefile an erster Stelle, damit sie vom FS5 besonders schnell gefunden werden kann.

Wenn Sie im FS5-Menü "Scenery" im "Library"-Untermenü den Eintrag "world.vis" angewählt haben, also auf die Standardszenerie von Microsoft® zugreifen, können Sie Ihre eigene Szenerie dadurch sichtbar machen, daß Sie ihr die Ident-Nummer 1 geben. (Andere erlaubte Werte sind auch 3, 4 und 5. Warum dies so ist, erfahren Sie im Kapitel "Die FS5-Szeneriebibliothek".) Zum Setzen der Ident-Nummer können Sie den Befehl "ID" benutzen, gefolgt von der gewünschten Ident-Nummer, also z.B. ID 1

Der zweite Befehl hat die Aufgabe, dem FS5DesignerTool am Ende eines Quelltextes mitzuteilen, daß er seine Übersetzungsarbeit beenden kann. Er heißt schlicht END

Verwenden Sie diesen Befehl bitte immer am Ende Ihrer Quelltexte.

Als nächstes wollen wir uns den verschiedenen Koordinatendarstellungen widmen, die das FS5DesignerTool unterstützt.

2.3 Koordinatendarstellungen

Der FS5 verwendet Szeneriedaten immer nur an den Stellen, für die sie definiert wurden. Hierfür verwendet der FS5 verschiedene Koordinatendarstellungen, die alle dazu dienen, ein Objekt im dreidimensionalen Raum zu positionieren. Die drei Dimensionen sind Latitude (entspricht dem geläufigen Breitengrad), Longitude (Längengrad) und Altitude (Höhe über Meeresspiegel).

Der FS5 und das FS5DesignerTool kennen mehrere Arten von dreidimensionalen Koordinaten :

- absolute Koordinate
- relative Koordinate
- Blockkoordinate

2.3.1 Absolute Koordinate

Am einfachsten ist die absolute Koordinatendarstellung zu erläutern. Dabei wird ein Punkt auf der Landkarte (mit Längen- und Breitengrad sowie einem Höhenwert) dargestellt als z.B.

(N51,36,55) , (E8,37,2) , 228

Dies sind die Koordinaten des Flughafens Paderborn-Lippstadt, nämlich 51° 36' 55" nördlicher Breite, 8° 37' 2" östlicher Länge und 228 m über dem Meeresspiegel. Im FS5DesignerTool werden Latitude- und Longitude-Wert also in Blöcken zu drei Werten gruppiert als

([[N|S] | [E|W]] GG, MM, SS.SSSS)

Die Kürzel N, S, E, W stehen für die Himmelsrichtungen und können auch weggelassen werden. Standardmäßig sind "N" für Latitude und "E" für Longitude voreingestellt. Bitte beachten Sie auch, daß die Sekundenkomponente als einzige maximal vier Nachkommastellen enthalten darf.

2.3.2 Relative Koordinate

Ähnlich einfach verhält es sich mit den relativen Koordinaten. Hier wird in kleineren Dimensionen gedacht, z.B. wird nur ein Bereich von 2 km x 2 km betrachtet. In dieser Darstellung beziehen sich die Koordinaten nicht auf

ein Grad-Minute-Sekunde-System, sondern auf einen Referenzpunkt, der innerhalb unseres 2 km x 2 km-Gebietes von uns selbst definiert wurde. Hier werden einfach die Latitude-, Longitude- und Altitude-Abweichung in m, bezogen auf unseren Bezugspunkt angegeben.

Als Beispiel betrachten wir einen Befehl, der eine Linie zieht vom Bezugspunkt zu einem neuen Punkt:

```
LINE_TO 300,150,0
```

Hier wird eine Linie gezogen, ausgehend vom Bezugspunkt 300 m in Richtung Norden (entspricht einer Latitude-Verschiebung), 150 m in Richtung Osten (Longitude-Verschiebung) und 0 m in die Höhe. (Das vollständige Beispiel zu diesem Linienbefehl finden Sie im Kapitel "einfache Operationen" im Abschnitt "Linien und Straßen".)

Da die Komponenten einer relativen Koordinate nur 16 Bit lang sind, dürfen die verwendeten Zahlen (oder numerischen Ausdrücke) nicht kleiner als -32768 und nicht größer als 32767 sein.

2.3.3 Blockkoordinate

Die komplizierteste Darstellungsform ist die Blockkoordinate. Da diese aber ausschließlich zum Erzeugen von Bodenstrukturen benötigt werden, können Sie dieses Kapitel erst einmal "überlesen", bis Sie selber Bodenstrukturen bauen wollen.

Die Koordinaten dieses Typs sind in sechs Unterklassen unterteilt, deren Strukturen aber fast identisch sind. Einziges Unterscheidungsmerkmal sind die Größenordnungen, in denen sie gemessen werden. Die Typ 1-Koordinaten sind am größten, die Typ 6-Koordinaten die genauesten. Eine Einheit (1 "Block") einer Koordinate entspricht einer Grad-Minute-Sekunde-Änderung. Hier sind die genauen Zahlen:

Typ	Verschiebung	Latitude-Wertebereich	Longitude-Wertebereich
1	1° 24' 32"	-66.. 66	0.. 255
2	42' 16"	-133.. 133	0.. 511
3	21' 8"	-267.. 267	0.. 1023
4	10' 34"	-535.. 535	0.. 2047
5	5' 17"	-1071.. 1071	0.. 4095
6	2' 38.5"	-2143.. 2143	0.. 8191

Bei keinem der Typen 1 bis 6 ist es also möglich, einen ganz bestimmten Punkt auf der Erde zu referenzieren, sondern vielmehr entspricht ein bestimmter Koordinatenwert einer quadratischen Fläche mit der aus der Tabelle ersichtlichen Seitenlänge.

Dabei entspricht die absolute Koordinate (0,0,0),(0,0,0),0 (Äquator auf dem 0. Längengrad) der Blockkoordinate 0,0 vom Typ 1. Die Blockkoordinate besteht nur aus zwei Komponenten (für Latitude und Longitude), weil sie keine exakte Höhe beschreibt, sondern nur für Bodendefinitionen benutzt wird. Blockkoordinaten können nur für Latitude-Werte definiert werden, die kleiner sind als 68° 34'. In der Tabelle oben sind für alle Typen (1 bis 6) die gültigen Latitude-Werte von Blockkoordinaten angegeben.

Der Longitude-Wertebereich deckt das Intervall 0°..360° ab. Bei Typ 0 ist dies ein Wertebereich von 0 bis 255, bei Typ 1 liegen gültige Longitude-Werte im Bereich 0..511, d.h. der Wertebereich ist doppelt so groß. In einen Block des Typs 1 passen ja auch vier Blöcke des Typs 2. Dasselbe gilt für die Beziehungen aller anderen aufeinanderfolgenden Typen. Der Wertebereich für Longitude-Werte des Typs 6 geht damit von 0 bis 8191.

2.4 Numerische Ausdrücke

Das FS5DesignerTool bietet in der vorliegenden Version die Verwendung einfacher numerischer Ausdrücke. Sämtliche Zahlen-Parameter von Befehlen dürfen über die Operationen "+", "-", "*", "/" verknüpft werden, also etwa in

```
LINE_TO 26*10+40,150,0
```

Eine explizite Vorrangregelung wie etwa "Punktrechnung vor Strichrechnung" gibt es hier jedoch nicht. Die Ausdrücke werden vielmehr von links nach rechts ausgewertet, d.h. der Ausdruck

```
40+26*10
```

ergibt nicht etwa

```
40 + (26*10) = 40 + 260 = 300,
```

sondern wir erhalten

```
(40+26) * 10 = 66 * 10 = 660.
```

Bitte beachten Sie beim Verwenden von numerischen Ausdrücken immer die Möglichkeit eines Überlaufs; viele Operanden von FS5DesignerTool-Befehlen dürfen beispielsweise nur 16 Bit lang sein. Bei der Multiplikation zweier 16-Bit-Zahlen kann das Ergebnis aber bis zu 32 Bit lang werden. Das FS5DesignerTool macht keinen Überlaufstest.

2.5 Strings

Das FS5DesignerTool unterstützt die Verwendung von einfachen Strings, wie sich aus einfachen kommandoorientierten Programmiersprachen bekannt sein dürften. Ein String ist eine in Anführungszeichen (") gesetzter Text, der selbst keine Anführungszeichen enthalten darf. Ein Beispielstring wäre also

```
"Dies ist ein String."
```

Strings werden uns helfen, die Lesbarkeit der Quelltexte erträglich zu halten, indem wir etwa Sprungziele oder den Beginn von Unterprogrammen mit aussagekräftigen Markierungen, sogenannten "Labels" versehen werden.

Eine Beispielsequenz wäre etwa

```
CALL "Zeichne Haus"
JUMP "Weiter"
LABEL "Zeichne Haus"
' ... hier müßte das Unterprogramm zum Zeichnen des Hauses stehen
RETURN
LABEL "Weiter"
' ... und hier geht's weiter
```

2.6 Kommentare

Um die Übersicht über den eigenen Quelltext zu erhalten, erlaubt das FS5DesignerTool in allen Eingabetexten auch Kommentare. Diese werden mit einem Hochkomma (') eingeleitet und gelten bis zum Ende der aktuellen Zeile. Ein Beispiel wäre also:

```
LINE_TO 300,150,0 'dies ist das Beispiel von eben
```

Ebenso werden in Befehlsfolgen Leerzeilen, TABs und Leerzeichen ("Blanks") überlesen. Sie können Ihren Programmtext also selbstverständlich einrücken und mit Leerzeilen versehen.

3. Flugnavigation

Als erste konkrete Szenarietypen wollen wir die verschiedenen Möglichkeiten der Flugnavigation kennenlernen, die das FS5DesignerTool durch einfache Befehle unterstützt:

- Drehfunkfeuer (VOR)
- Instrumentenlandesysteme (ILS)
- ungerichtete Funkfeuer (NDB)
- Voreinflugzeichen (OUTER_MARKER)
- Haupteinflugzeichen (MIDDLE_MARKER)
- Platzeinflugzeichen (INNER_MARKER)

Zu diesem Zweck richten wir uns nun eine "Basis"-Datei ein, die wir "nav.fst" nennen wollen. Wir werden diese in den Abschnitten dieses Kapitels nach und nach ergänzen. Sie finden diese Datei auch unter den mitgelieferten Beispielen auf der Programmdiskette:

```
'Dies ist eine Beispieldatei
SCENERY 18,0,0,0,0,"Aerosoft Welt",
        "Dies ist eine Beispielszenerie zum Austesten des FS5DesignerTools."
'Identifiziere Szenerie
ID 18
'hier steht der eigentliche Programmcode
END
```

Den SCENERY-Befehl haben wir noch nicht besprochen; wir benutzen ihn hier nur um zu vermeiden, daß sich unsere Szeneriedaten mit anderen Szenerien überschneiden.

Ebenfalls im Lieferumfang befindet sich eine Hilfsdatei namens "f.bat". Um diese kleine Hilfe zu benutzen, geben Sie einfach "f nav" ein. Sie können dann ihren Quelltext eingeben und brauchen anschließend nichts weiter zu tun als den Editor zu verlassen; Ihr "Programm" wird dann übersetzt und, wenn kein Fehler aufgetreten ist, wird der FS5 mit Ihren Daten gestartet.

Beim Starten des FS5 mit der eigenen Szenerie müssen Sie dann nur noch im Menü "Scenery" unter "Scenery Library" den Eintrag "Aerosoft Welt" anwählen. Am besten, Sie speichern die so erhaltene Situation dann ab und konfigurieren Ihren FS5 auch so um, daß nach dem Start automatisch diese Situation geladen wird.

Als Koordinaten können Sie dann schon einmal N51.36.55 und E8.37.2 im FS5 einstellen. Dies ist die Position des Flughafens Paderborn-Lippstadt. Wir wollen an dieser Position verschiedene Navigationshilfen erläutern.

Wenn Sie übrigens die Probe auf's Exempel machen wollen und "nav.fst" so übersetzen, wie es ist, werden Sie nach Starten des FS5, Einstellen auf obige Koordinaten und Anwahl der Aerosoft-Szenerie eine quasi "leere" Szenerie erhalten, die aus nur einer Bodenfarbe besteht.

3.1 VOR

Der Flughafen Paderborn-Lippstadt verfügt über ein eigenes Drehfunkfeuer. Seine Koordinaten sind N51.36.55, E8.37.2 bei 213 m Höhe. Seine Frequenz ist 108.5 MHz, bei einer Reichweite von ca. 60 km. Die magnetische Abweichung beträgt 1°. Die Kennung des Funkfeuers lautet "PAD", der volle Name ist "Paderborn/Lippstadt". Weiter wissen wir, daß das Drehfunkfeuer mit einer Einrichtung zur Distanzmessung, einem sogenannten DME versehen ist.

Wenn wir dieses Funkfeuer im FS5DesignerTool programmieren wollen, dann sieht das wie folgt aus:

```
VOR 108.5, 60000, 1, (N51,36,55), (E8,37,2), 213, "PAD", "Paderborn-Lippstadt", 1
```

Im ersten Parameter steht also die Frequenz, gefolgt von der Reichweite des Funkfeuers. Dann folgen magnetische Abweichung in Grad, Latitude und Longitude als absolute Koordinaten, die Höhe in m, die Kennung, der Name und schließlich die Angabe, daß ein DME vorhanden ist.

Versuchen Sie einmal, diesen Befehl an der dafür freigehaltenen Stelle in unserem "nav.fst" einzusetzen und sehen Sie sich das Ergebnis im FS5 an. Wenn der FS5 korrekt startet, stellen Sie im NAV1-Fenster doch einmal die Frequenz 108.5 MHz ein. Im DME1-Fenster sollte jetzt eine Distanz in NM auftauchen - vorausgesetzt, Sie sind nicht zu weit vom DME entfernt.

Im FS5-Menü "NAV / COM" können Sie sich unter "Navigation Radio" überzeugen, daß Ihr VOR richtig erkannt wird. Unter der Frequenz sollten Sie nun den Namen des VORs und das Kürzel finden.

Allgemein lautet die Syntax bei VORs:

```
VOR <Frequenz in MHz>, <Reichweite in m>, <magnetische Abweichung in Grad>,  
  <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>,  
  "<ID, Länge <=5>","<Name, Länge <=24>","<DME aus {0,1}>"
```

Beachten Sie bitte, daß der ID-String nicht länger als 5 Zeichen sein darf; bei längeren Eingaben bricht das FS5DesignerTool seine Arbeit ab. Ebenso beim Funkfeuer-Namen, der aus maximal 24 Zeichen bestehen darf.

3.2 ILS

Als nächstes wollen wir uns ein Instrumentenlandesystem (ILS) anschauen, weil dies in seiner Struktur dem Drehfunkfeuer (VOR) am nächsten kommt. Wir werden daher einfach so tun, als sei das Drehfunkfeuer, das wir im letzten Abschnitt definiert haben, ein komplettes Instrumentenlandesystem (ILS).

Die Hauptkomponenten eines ILS sind ein hochempfindliches VOR, ein Gleitwegsender und sogenannte Markierungsfunkfeuer. Wir werden im ILS-Befehl des FS5DesignerTool aber nur das VOR und den Gleitwegsender definieren; für die Markierungsfunkfeuer existieren eigene Befehle, auf die wir im Abschnitt "IMO Marker" kommen werden.

Wir nehmen einmal an, das ILS-Drehfunkfeuer sei dasjenige, das wir im VOR-Befehl des letzten Abschnitts definiert haben. Bleibt also nur noch die Beschreibung des Gleitwegsenders. Dafür nehmen wir der Einfachheit halber an, daß der Gleitwegsender die selben Koordinaten aufweist wie das Drehfunkfeuer. Zusätzlich wird der Gleitwegsender charakterisiert durch einen Winkel. Wir wollen ihn mit 3° annehmen. Schließlich benötigen wir noch einen Anflugkurs in Grad; wir setzen diesen auf die Orientierung der Landebahn, also 239°. Wir erhalten dann

```
ILS 108.5, 60000, 1, (N51,36,55), (E8,37,2), 213,  
  "PAD", "Paderborn-Lippstadt", 193, 239, (N51,36,55), (E8,37,2), 213, 3
```

Es wird deutlich, daß die Syntax des ILS-Befehls der des VOR-Befehls aus dem letzten Abschnitt ähnelt. Der 9. Parameter "Flags" hat hier aber eine andere Bedeutung: Es kann nicht nur spezifiziert werden, ob ein DME vorhanden ist (Wertigkeit "1" für DME), sondern auch:

Wert	Bedeutung
1	DME
64	Gleitwegsender
128	Empfindlichkeit

Um mehrere Features zu erhalten, sind die jeweiligen Wertigkeiten aufzuaddieren und nach "Flags" zu schreiben.

In den letzten fünf Parametern sind dann die Informationen über den Gleitwegsender abgelegt. Beachten Sie hierbei bitte, daß der Winkel des Gleitwegsenders nur innerhalb eines festen Wertebereichs spezifiziert werden darf, nämlich dem Intervall [-14.3275°... +14.3275°]. Durch den kleineren Wertebereich wird eine genauere interne Darstellung des Winkels im FS5 erreicht.

Insgesamt lautet das Befehlsformat also:

```
ILS <Frequenz in MHz>, <Reichweite in m>, <magnetische Abweichung in grad>,  
  <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>,  
  "<ID, Länge <=5>","<Name, Länge <=24>","<Flags aus [0..255]>,"  
  <Anflugkurs in grad>,"  
  <Gleitwegsender Latitude in (G,M,S)>," <Gleitwegsender Longitude in (G,M,S)>,"  
  <Gleitwegsender Altitude in m>,"  
  <Gleitwegsender-Winkel in grad aus [-14.3275°... +14.3275°]>
```

3.3 NDB

Die Definition von ungerichteten Funkfeuern (NDBs) läßt sich mit dem FS5DesignerTool leicht bewerkstelligen: Alles, was wir dafür benötigen, sind die Position, ID und Name, sowie Frequenz und Reichweite.

Beginnen wir mit der Frequenz (Parameter 1). Diese ist in Kilohertz (kHz) anzugeben und darf auch Nachkommastellen aufweisen. Allerdings sind nur zwei Nachkommawerte erlaubt, nämlich die ".0" und die ".5". Wir nehmen für unser Beispiel einmal als Frequenz 218.0 kHz an.

Die Reichweite des NDBs ist in Einheiten von 2048 Metern anzugeben, d.h. wenn unser NDB beispielsweise eine Reichweite von 30 km aufweist, so geben wir $30000 : 2048 = \text{ca. } 15$ an.

Die Koordinatenwerte eines NDBs sind - wie bei VOR und ILS - als Absolutkoordinaten anzugeben. Und, ID und Name dürfen nur maximal 5 bzw. 24 Zeichen lang sein.

Ein Beispiel-NDB beim Flughafen Paderborn-Lippstadt könnte also lauten:

NDB 218,255, (51,40,0) , (8,37,2) , 228, "NDB1", "Test-NDB for Paderborn"

3.4 IMO Marker

Als letzte Navigationshilfen wollen wir die verschiedenen Einflugzeichen ("Marker") kennenlernen. Wir unterscheiden Voreinflugzeichen (OUTER_MARKER), Haupteinflugzeichen (MIDDLE_MARKER) und Platzeinflugzeichen (INNER_MARKER). Sie kennen diese Elemente vielleicht aus dem FS5-Cockpit; diese leuchten immer dann auf, wenn Sie das betreffende Einflugzeichen überfliegen.

Alles, was Sie zum Setzen von Einflugzeichen benötigen, sind die dazugehörigen Absolutkoordinaten. Wir setzen jetzt einmal beispielhaft alle drei Einflugzeichen, wieder in der Nähe des Flughafens Paderborn-Lippstadt.

OUTER_MARKER (51,36,0) , (8,37,2) , 228

MIDDLE_MARKER (51,36,53) , (8,37,2) , 228

INNER_MARKER (51,36,35) , (8,37,2) , 228

4. Einfache Grafikoperationen

Wir sind jetzt beim Kern des Szeneriebaus für den FS5 angelangt. Sämtliche Objekte wie Gebäude, Städte, Flüsse und einige Berge werden mithilfe der sogenannten primitiven Grafikoperationen erstellt, begleitet durch einige Befehle zur Unterstützung von Ablaufstrukturen wie Sprünge, Unterprogramme oder Zeitschleifen.

Doch auch hier sind wieder einige grundlegende Konzepte notwendig, bevor wir sichtbare Ergebnisse erzielen können. Wir wollen diese aber so kurz wie möglich und so lang wie nötig im folgenden Abschnitt "Sichtbarkeitsbereiche" abhandeln.

4.1 Sichtbarkeitsbereiche

Die Szenerien für den FS5 decken häufig weit größere Gebiete ab, als auf einmal auf dem Bildschirm dargestellt werden kann. Abhängig vom augenblicklichen Ort ist nur ein kleiner Ausschnitt der gerade geladenen Szenerie sichtbar.

Wir sprechen hier von sogenannten Sichtbarkeitsbereichen. Der FS5 erwartet, daß alle Szenerieinformationen bereits "vorsortiert" sind, d.h. Daten, die im selben Sichtbarkeitsbereich liegen, sollen im gleichen Szenerieabschnitt abgelegt werden.

Wir können uns einen Sichtbarkeitsbereich beispielsweise als einen 2 km x 2 km-Ausschnitt vorstellen, in dem Objekte wie Straßen, Häuser, Flüsse usw. definiert werden. Natürlich sind auch größere "Ausschnitte" möglich, die dann aber entsprechend "gröber" ausfallen.

Der FS5 erwartet, daß die unterschiedlichen Sichtbarkeitsbereiche in einer festgelegten Art abgelegt werden. Am Anfang jedes Sichtbarkeitsbereichs steht ein Verweis auf den nachfolgenden Sichtbarkeitsbereich. Dieses Konzept ermöglicht es dem FS5, schnell die Daten zu finden, die dem jeweiligen Sichtbarkeitsbereich entsprechen, das heißt, mit dieser Maßnahme wird der Zugriff auf die - teilweise recht komplexen - Szeneriedaten beschleunigt.

Als Benutzer des FS5DesignerTools brauchen Sie sich jetzt aber keine "grauen Haare" wachsen zu lassen. Die meiste Arbeit wird Ihnen vom FS5DesignerTool abgenommen. Es genügt zu wissen, wie ein Sichtbarkeitsbereich definiert wird. Hierfür steht der Befehl AREA_TEST zur Verfügung, der einen solchen Sichtbarkeitsbereich durch Angabe einer absoluten Koordinate (mit Latitude- und Longitude-Wert) sowie eines "Reichweite"-Wertes festlegt.

Wir wollen einen solchen Befehl einmal für den Flughafen Paderborn-Lippstadt angeben:

```
AREA_TEST (N51,36,55), (E8,37,2), 255
```

Die ersten zwei Parameter beschreiben die Latitude- und Longitude-Position des Sichtbarkeitsbereiches. Im letzten Parameter haben wir einfach die maximal mögliche "Reichweite" eingesetzt, nämlich 255. Allgemein dürfen hier Werte von 1 bis 255 stehen.

Nun ist es aber noch nicht mit der Definition des Sichtbarkeitsbereiches getan, um beispielsweise Häuser zu bauen oder Straßen anzulegen. Zusätzlich verlangt der FS5 noch die Definition eines sogenannten "Referenzpunktes". Dieser Punkt stellt sozusagen den Nullpunkt eines lokalen Koordinatensystems für Ihren Sichtbarkeitsbereich dar.

Haben Sie erst einmal einen Referenzpunkt definiert, können Sie (fast) alle nachfolgenden einfachen Grafikoperationen in den einfacheren relativen Koordinaten eingeben, das heißt, in 1m-Einheiten. Diese Darstellung ist zum Bau einfacher Objekte sicher geeigneter als die Angabe von absoluten Koordinaten mit Grad / Minute und Sekunde, denn, wer weiß schon, wieviel beispielsweise eine Häuserwand von 10 m in Sekunden ausgedrückt ergibt?

Um einen Bezugspunkt an der selben Koordinate wie im AREA_TEST zu vereinbaren, geben Sie also - nach dem AREA_TEST-Befehl - folgenden Befehl an:

```
REF_POINT (N51,36,55), (E8,37,2), 0, 1, 1
```

Da wir praktisch noch keine Bodenstrukturen haben, müssen wir in diesem Beispiel noch von 0 m Bodenhöhe ausgehen; deshalb steht als dritter Parameter (Höhe über NN) eine "0". Dies ist die Standardeinstellung für Bodenflächen. Tatsächlich liegt der Flughafen Paderborn-Lippstadt in 699 Fuß Höhe.

Es bleiben noch die Parameter 4 und 5 zu klären. Diese legen quasi die Basiseinheit "1 m" für alle nachfolgenden einfachen Operationen fest. Allgemein stellt der 4. Parameter einen Streckungsfaktor dar, während der 5. Parameter ein Streckungsdivisor ist. Die Werte "1, 1", stehen etwa für die Basiseinheit "1 m". Verein-

baren wir nun "2, 1", so werden (fast) alle Grafikobjekte mit doppeltem Maßstab gezeichnet. Sie können hiermit also leicht auch größere Objekte erzeugen, indem Sie einfach den Streckungsfaktor vergrößern.

Bei unseren umfangreichen Tests konnten wir feststellen, daß die Parameter 4 und 5 keine linearen Streckungsfaktoren darstellen - ein Faktor von zehn entspricht also keinesfalls einer zehnfach vergrößerten Darstellung; hier gilt es daher zu experimentieren.

Die meisten einfachen Operationen können ohne Angabe eines Bezugspunktes erst gar nicht ausgeführt werden. Der FS5 "überliest" die entsprechenden Codesequenzen dann einfach. **Achten Sie also immer darauf, gleich nach dem AREA_TEST einen REF_POINT zu vereinbaren.**

Lediglich der RUNWAY-Befehl (wird im Abschnitt "Flughäfen und Landebahnen" besprochen) sowie die Sprung- und Unterprogrammbefehle (z.B. "JUMP", "CALL") verlangen nicht explizit die Vorgabe eines Referenzpunktes.

4.2 Farben

Bevor wir ein primitives Grafikobjekt auf den Bildschirm bringen wollen, müssen wir ihm entweder ein Muster oder aber eine Farbe geben. In den meisten Fällen können wir dies erreichen, indem wir einen Farb-Befehl vom Typ COLOUR unmittelbar vor der Grafikoperation verwenden.

Wir unterscheiden drei COLOUR-Befehle. SHADE_COLOUR wird beim Spezialbefehl "SHADED_POLYGON" eingesetzt, mit dem sich Objekte erzeugen lassen, die Farbverläufe aufweisen. LINE_COLOUR dient, wie der Name schon nahelegt, dem Setzen von Farben für zu zeichnende Linien. GROUND_COLOUR wird verwendet, wenn Bodenflächen in einer bestimmten Farbe gezeichnet werden sollen (z.B. Umrisse von Städten), oder zum Einfärben von Straßen.

Da es in den meisten Fällen gar nicht notwendig ist anzugeben, von welcher Art das nachfolgende zu zeichnende Objekt ist, wurde zusätzlich der Befehl COLOUR eingeführt. Dieser hat dieselbe Wirkung wie der Befehl GROUND_COLOUR. Die Syntax der Befehle lautet:

```
COLOUR <Nummer der Farbe>
SHADE_COLOUR <Nummer der Farbe>
LINE_COLOUR <Nummer der Farbe>
GROUND_COLOUR <Nummer der Farbe>
```

Der Wertebereich für Farben ist das Intervall [0..255].

Die meisten existierenden Farben werden der Tageszeit angepaßt; nur wenige Farben verhalten sich bei Tag und Nacht gleich. Da es nicht möglich ist, alle Farben mit ihren möglichen Abstufungen verbindlich aufzulisten, befindet sich im Lieferumfang ein Beispielprogramm namens "colours.fst". Näheres dazu im Anhang.

Standardmäßig benutzt der FS5 eine Farbpalette, die als Datei im FS5-Directory "texture" steht; sie trägt den Namen "fs5.pal". Mit einem speziellen Befehl, "PALETTE", haben Sie jedoch die Möglichkeit, eine besondere Farbpalette anzuwählen. Für die unmittelbar nachfolgenden primitiven Grafikbefehle werden dann die Farben der angewählten Farbpalette verwendet.

```
PALETTE "meigs.pal", 0
```

Der zweite Parameter wurde zwar bereits für den "PALETTE"-Befehl implementiert, seine genaue Wirkung ist aber noch nicht klar. Setzen Sie diesen Parameter einfach auf "0". Achten Sie bei der Verwendung dieses Befehls bitte darauf, daß die angewählte Farbpalette im FS5-Directory "texture" steht; der FS5 bricht seine Arbeit sonst ab.

Abschließend wollen wir noch einen Befehl zum Setzen der Farb-Intensität erwähnen, "COLOUR_BRIGHTNESS". Er hat nur einen Parameter, mit Wertebereich [0..65535]. Auch seine Wirkung konnte noch nicht genau erforscht werden.

4.3 Linien und Straßen

Wir wollen uns nun erstmals der Vereinbarung von Objekten widmen, die auch auf dem FS5-Bildschirm sofort ersichtlich sind. Wir wollen uns nämlich mit der Erzeugung von einfachen Straßen, Taxiways und einfachen Linien beschäftigen.

Der FS5 stellt bereits zwei verschiedene Typen von Straßen zur Verfügung, ROAD1 und ROAD2. Zu jeder Straße können wir Breite, Länge sowie Start- und Endpunkt definieren. Wir betrachten zuerst beispielhaft Straßen vom Typ ROAD1.

Im Befehl ROAD_MOVE_TO1 legen wir den Startpunkt einer Straße sowie ihre Breite fest:

```
ROAD_MOVE_TO1 0,0,0, 30
```

Mit diesem Befehl definieren wir den Beginn unserer Straße genau auf dem vorher definierten Bezugspunkt und geben ihr die Breite 30 m. Gezeichnet wird hier noch nichts.

Wenn wir ein Stück Straße zeichnen wollen, müssen wir vorher eine Farbe definieren und können dann mit ROAD_LINE_TO1 ausgehend vom vorher definierten Startpunkt (bzw. vom zuletzt mit ROAD_MOVE_TO1 gezeichneten Endpunkt des letzten Straßenstücks) eine Verbindung zu einem festzulegenden neuen Endpunkt zeichnen. Hier einmal das komplette Codestück:

```
AREA_TEST (N51,36,55), (E8,37,2),255
REF_POINT (N51,36,55), (E8,37,2),0,1,1
GROUND_COLOUR 10
ROAD_MOVE_TO1 0,0,0,30
ROAD_LINE_TO1 1000,0,0
ROAD_LINE_TO1 1000,1000,0
```

Mit dem ersten ROAD_LINE_TO1-Befehl zeichnen wir ein Straßenstück vom Punkt 0,0,0 zum Punkt 1000,0,0, d.h. wir bewegen uns um 1000 m = 1 km in Richtung Norden. Der zweite ROAD_LINE_TO1-Befehl legt ein Straßenstück fest zwischen den Punkten 1000,0,0 und 1000,1000,0, d.h. wir bewegen uns um 1 km in Richtung Osten.

Wenn wir an einer neuen Stelle ansetzen wollen, ohne eine Straße zu zeichnen, können wir auch jederzeit innerhalb einer solchen Sequenz den ROAD_MOVE_TO1-Befehl benutzen. Auf diese Weise können wir schnell komplexe Straßengebilde erzeugen. Sie können auch schon an diesem Beispiel einmal ausprobieren, wie Straßen nachts dargestellt werden: An Beginn und Ende jedes Straßenstücks werden dann kleine gelbe Punkte (Straßenlichter) gesetzt, damit die Straße auch nachts sichtbar ist.

Haben Sie auch gemerkt, was passiert, wenn Sie sich von der Straße entfernen bzw. sich ihr nähern? Sobald Ihr Abstand zur Straße einen bestimmten Wert (ca. 1000 m) überschritten hat, wird die Straße nur noch als dünne Linie gezeichnet; erst, wenn Sie diese Distanz unterschreiten, wird wieder auf eine breite Linie umgeschaltet.

Nach unserem derzeitigen Kenntnisstand läßt sich diese manchmal unerwünschte "Eigenart" des FS5 auch dadurch umgehen, daß wir Straßen anstatt mit den vordefinierten ROAD-Befehlen "von Hand" erzeugen, indem wir sie als Polygone definieren. Wie man solche Polygone erzeugt, wird im Abschnitt "Flächen - Mal gefüllt, mal farbig" beschrieben. Auf alle Fälle ist der Weg über Polygone weitaus aufwendiger, weil Sie viel mehr Zwischenberechnungen anstellen müssen, als bei den einfachen ROAD-Befehlen.

Haben Sie erst einmal das Prinzip verstanden, nach dem Straßen vom Typ 1 gezeichnet werden, können Sie auch schon Straßen des Typs 2 und Taxiways definieren. Hierfür existieren die äquivalenten Befehle ROAD_MOVE_TO2, ROAD_LINE_TO2 sowie TAXI_MOVE_TO und TAXI_LINE_TO. Die Parameter sind bei den MOVE-Befehlen genau wie bei ROAD_MOVE_TO1 dargestellt; dasselbe gilt für die Typ-LINE-Befehle.

Wir wollen nun eine andere Methode kennenlernen, um Straßen zu definieren. Mit den bisher beschriebenen Befehlen werden Straßen ja über Koordinatenpunkte beschrieben; das ist nicht immer der einfachere Weg. Wir könnten Straßen ja auch dadurch definieren, daß wir, ausgehend von einem Startpunkt, jedes Teilstück durch seine Länge und Orientierung beschreiben. Dann können wir uns einfach eine Autokarte vornehmen und brauchen eigentlich nur noch ein Lineal und einen Winkelmesser.

Genau dieses Konzept ist in den Befehlen ROAD_DRAW1, ROAD_DRAW2 und TAXI_DRAW verwirklicht. Diese entsprechen wieder den drei bereits bekannten Straßentypen ROAD1, ROAD2 und TAXI. Unser letztes Beispiel reduziert sich dann auf nur noch 4 Befehle:

```
AREA_TEST (N51,36,55), (E8,37,2),255
REF_POINT (N51,36,55), (E8,37,2),0,1,1
GROUND_COLOUR 10
ROAD_DRAW1 0,0,0,30,2,
           1000,0,
           1000,90
```

In den ersten vier Parametern legen wir die Koordinaten des Startpunktes und die Straßenbreite fest. Dann folgt die Anzahl der Teilstücke. Für jedes Teilstück definieren wir dann noch dessen Länge in m und die Orientierung in Grad. 0° entspricht in unserer Darstellung der Richtung "Norden", 90° "Osten", 180° "Süden" und 270° "Westen". Die allgemeine Syntax lautet:

```
ROAD_DRAW1 <rel.Startlat.in m>, <rel.Startlong.in m>, <rel.Startalt.in m>,
           <Breite der Straße in m>, <Anzahl n der Teilstücke>,
           <Länge des 1.Teilstücks in m>, <Richtung des 1.Teilstücks>,>
```


...
<Länge des n.Teilstücks in m>, <Richtung des n.Teilstücks>

Wohlgemerkt, dieser Befehl erzeugt denselben Straßentyp wie die oben angesprochenen Befehle ROAD_MOVE_TO1 und ROAD_LINE_TO1. ROAD_DRAW1 ist nur eine andere Art der Vereinbarung von Straßen. Wir haben es hier auch erstmals mit einem Befehl zu tun, der eine variable Anzahl von Parametern erlaubt (hier: abhängig von der Anzahl der Straßenstücke).

Neben Straßen haben Sie beim FS5DesignerTool auch die Möglichkeit, Linien zu definieren, die die Orientierung vereinfachen können. Dafür kennen wir wieder zwei Methoden.

Die erste Methode ähnelt der Vereinbarung von Straßen, nur daß Sie keine Linienbreite mehr angeben können. Mit MOVE_TO legen wir den Startpunkt einer Linie fest, mit LINE_TO läßt sich eine Linie zeichnen, hin zu einem Endpunkt, den Sie selber definieren.

Wir wollen einmal eine 1000 m aufragende rote Linie zeichnen, anstatt einer Straße:

```
AREA_TEST (N51,36,55), (E8,37,2),255
REF_POINT (N51,36,55), (E8,37,2),0,1,1
COLOUR 15
MOVE_TO 0,0,0
LINE_TO 0,0,1000
```

Diese Befehle sind den bekannten ROAD- und TAXI-Befehlen sehr nah verwandt.

Die zweite Methode der Vereinbarung von Linien funktioniert anders. Bei diesem Verfahren werden vor dem Zeichnen von Linien erst einmal Listen von Punkten definiert. Die so definierten Punkte sind aber nichts anderes als die direkt vorgegebenen Koordinatenwerte aus den bisher bekannten Befehlen. Wir betrachten einmal die Vereinbarung von solchen Punktlisten mit dem FS5DesignerTool.

```
AREA_TEST (N51,36,55), (E8,37,2),255
REF_POINT (N51,36,55), (E8,37,2),0,1,1
DEFINE_POINTS 4,0,0,0,0,1000,0,0,1000,1000,0,0,1000,0
LINE_COLOUR 15 'rote Linien
MOVE_TO_POINT 0
LINE_TO_POINT 1
LINE_TO_POINT 2
LINE_TO_POINT 3
LINE_TO_POINT 4
```

Zum Definieren der Punktliste dient der Befehl DEFINE_POINTS. In ihm wird zunächst die Anzahl der zu definierenden Punkte festgelegt (erster Parameter), dann folgt der Startindex. Das ist der Index des ersten definierten Punktes, in unserem Fall also die "0". Die nachfolgenden Punkte tragen damit automatisch die Indices "1", "2" und "3". Anschließend folgen die Koordinatendaten der 4 Punkte in Latitude, Longitude und Altitude (immer bezogen auf den aktuellen Bezugspunkt).

Achten Sie bitte darauf, daß Punktedaten immer in dem Sichtbarkeitsbereich definiert werden müssen, in dem sie verwendet werden sollen. Das bedeutet unter anderem auch, daß Sie in einem anderen Sichtbarkeitsbereich wiederum Punkte definieren können, deren Indices bei "0" beginnen.

Die Vereinbarung von Punktelisten ist sogar so "lokal", daß Sie auch in einem Sichtbarkeitsbereich mehrmals Punkte mit denselben Indices definieren dürfen. Es werden dann immer die zuletzt gültigen Definitionen benutzt.

Nun können wir jeden Punkt über seine Nummer "adressieren", anstatt immer wieder die relativen Koordinaten anzugeben. Das ist eigentlich erst dann hilfreich, wenn bestimmte Punkte mehrmals verwendet werden. Das ist z.B. dann der Fall, wenn sich mehrere Linien in einem Punkt kreuzen.

Die Bedeutung der nachfolgenden Befehle dürfte offensichtlich sein: Wir bewegen unseren Zeichenzeiger auf den Startpunkt 0 und zeichnen von dort aus rote Linien zwischen den Punkten (0,1), (1,2), (2,3) und (3,0). Das erzeugte Objekt ist ein Quadrat.

Nun werden Sie sich vielleicht fragen, wofür solche Linien überhaupt zu gebrauchen sind beim Bau von FS5-Szenarien. Genaugenommen sind es nicht die Linien, um die es hier geht, sondern die Bereiche, die diese Linien eingrenzen können.

Wir können nämlich Linien als Begrenzungen von Bodenflächen benutzen, z.B. als Begrenzung eines Stadtrisses. Solche Objekte werden im FS5DesignerTool als "SURFACE", zu deutsch "Oberfläche", bezeichnet. Wir

wollen an unserem letzten Beispiel einmal demonstrieren, wie einfach es ist, die durch die 4 Linien definierte Fläche auszufüllen:

```
AREA_TEST (N51,36,55), (E8,37,2), 255
REF_POINT (N51,36,55), (E8,37,2), 0,1,1
DEFINE_POINTS 4,0,0,0,0,1000,0,0,1000,1000,0,0,1000,0
GROUND_COLOUR 50 ' graue Fläche
BEGIN_SURFACE
MOVE_TO_POINT 0
LINE_TO_POINT 1
LINE_TO_POINT 2
LINE_TO_POINT 3
LINE_TO_POINT 0
END_SURFACE
```

Wir haben hier nur um die MOVE_TO_POINT- / LINE_TO_POINT-Sequenz die Befehle "BEGIN_SURFACE" und "END_SURFACE" gesetzt und den COLOUR-Befehl in ein "GROUND_COLOUR" umgeändert. Das Ergebnis ist ein grau gefülltes Bodenpolygon.

Wenn Sie wollen, können Sie schon ein wenig experimentieren, indem Sie sich anders geformte Umrisse definieren. Auf diese Weise lassen sich komplexe Stadumrisse modellieren.

Wir wollen währenddessen noch zu einem letzten Linienbefehl kommen, der in seinen Möglichkeiten weitaus eingeschränkter ist, aber für schöne Effekte benutzt werden kann. Mit "DOTTED_LINE" lassen sich nämlich unterbrochene Linien zwischen zwei festen Punkten definieren.

Was Sie hierfür brauchen, sind lediglich der Start- und Endpunkt in relativen Koordinaten und die Anzahl der Teillinien, aus denen sich die Gesamtlinie zusammensetzen soll:

```
DOTTED_LINE <rel.Lat.P1>, <rel.Long.P1>, <rel.Alt.P1>,
<rel.Lat.P2>, <rel.Long.P2>, <rel.Alt.P2>,
<Anzahl der Teillinien für die Verbindung P1-P2>
```

Wenn wir also eine Linie von Punkt 0,0,0 zu Punkt 100,0,0 ziehen wollen, die aus 10 Teilstrichen besteht, müssen wir wie folgt codieren:

```
AREA_TEST (N51,36,55), (E8,37,2), 255
REF_POINT (N51,36,55), (E8,37,2), 0,1,1
LINE_COLOUR 15
DOTTED_LINE 0,0,0,100,0,0,10
```

4.4 Flächen - Mal gefüllt, mal farbig

Im Abschnitt über Straßen und Linien haben wir bereits eine Methode kennengelernt, um eine farbige Fläche zu erzeugen. Die Befehle BEGIN_SURFACE und END_SURFACE wurden benutzt, um einen Umriss zu definieren.

Zur Definition von Bodenpolygonen (z.B. Städten und Flüssen) ist diese Methode die einfachste und gleichzeitig beste. Sollen demgegenüber komplexere Objekte erzeugt werden, z.B. spezielle Gebäude (Beispiel: Flughafen-tower), so reicht eine SURFACE-Definition nicht mehr aus, weil dann beispielsweise bestimmte Probleme der Sichtbarkeit bzw. Unsichtbarkeit von einzelnen Wänden des Gebäudes hinzukommen. Um auch ein solches, komplexeres Polygon auf dem FS5-Bildschirm darstellen zu können, sind zusätzliche Informationen über das darzustellende Objekt notwendig.

Wir wollen zunächst einmal mit einem einfachen **quaderförmigen Gebäude** beginnen, das wir mit den FS5DesignerTool-Befehlen bauen werden. Anhand eines Beispiels läßt sich die Verwendung von Polygonen im FS5 am besten erklären.

Angenommen, das Gebäude sei 30 m lang, 40 m breit und 70 m hoch. Um ein Polygon zu zeichnen, müssen wir zunächst einmal die Begrenzungspunkte des Polygons definieren. In unserem Fall sind dies einfach sämtliche Ecken des Gebäudes, also insgesamt acht Punkte. Zur Vereinbarung der Punkte benutzen wir wieder den schon bekannten Befehl DEFINE_POINTS. (Den weiter erforderlichen AREA_TEST und das Setzen des REF_POINTS wollen wir jetzt nicht mehr in jedem Beispiel auflisten.)

```
DEFINE_POINTS 8,0,
0, 0, 0, 'Punkt links unten vorne
30, 0, 0, 'Punkt links unten hinten
30, 40, 0, 'Punkt rechts unten hinten
```


0, 40, 0,	'Punkt rechts unten vorne
0, 0, 70,	'Punkt links oben vorne
30, 0, 70,	'Punkt links oben hinten
30, 40, 70,	'Punkt rechts oben hinten
0, 40, 70	'Punkt rechts oben vorne

Der Vollständigkeit halber möchten wir hier noch einen zweiten "DEFINE_POINTS"-Befehl erwähnen, den das FS5DesignerTool auch zur Verfügung stellt: "DEFINE_ROTATED_POINTS". Hiermit können Sie alle Punkte gleich um einen bestimmten Winkel in x-, y- und z-Richtung gedreht definieren. Näheres dazu finden Sie im Anhang, in der alphabetischen Liste der FS5DesignerTool-Befehle.

Nun kommen wir zu den Seitenflächen des Gebäudes. Insgesamt sind dies fünf Stück, die sechste Seite (der Boden) brauchen wir nicht zu zeichnen, da diese nie sichtbar ist. Wir zeichnen jetzt das gesamte Objekt, indem wir es in seine Seitenflächen zerlegen. Jede Seitenfläche, jedes Polygon ist definiert durch seine begrenzenden Punkte und einen Vektor, der senkrecht auf dem Polygon stehen muß. Dieser Vektor setzt sich zusammen aus einer Latitude-, Longitude- und Altitude-Komponente und löst die bei dreidimensionalen Objekten (wie etwa Häusern) vorhandenen Sichtbarkeitsprobleme. Der Vektor "sagt" dem FS5 nämlich, welche Seite jeder Wand die Außenseite bzw. Innenseite der Wand ist.

Nun ist das Berechnen von solchen Vektoren per Hand eine fehlerträchtige Sache. Die Komponenten des Vektors lassen sich nicht "durch genaues Hinsehen" angeben, vor allem dann nicht, wenn das Gebäude nicht in Orientierung einer der Himmelsrichtungen vereinbart worden ist. Das FS5DesignerTool löst dieses Problem auf elegante Weise. Wenn bei der Eingabe der Polygondaten eine leicht nachvollziehbare Bedingung eingehalten wird, ist das FS5DesignerTool in der Lage, selbständig den korrekten Vektor zu bestimmen.

Die Bedingung an die Eingabe ist, daß die Punkte, die ein Polygon definieren, in einer Reihenfolge aufgerufen werden, die dem Uhrzeigersinn entspricht. Damit ist nämlich implizit festgelegt, welche Seite des Polygons "außen" liegt.

Sie können dann die fünf Seitenflächen aufrufen durch:

COLOUR 10	' Farbe hellbraun
POLYGON 4, 0,0,0, 0,4,7,3	' Fläche vorne
POLYGON 4, 0,0,0, 3,7,6,2	' Fläche rechts
POLYGON 4, 0,0,0, 2,6,5,1	' Fläche hinten
POLYGON 4, 0,0,0, 0,1,5,4	' Fläche links
POLYGON 4, 0,0,0, 4,5,6,7	' Fläche Dach

Dabei entspricht der erste Parameter der POLYGON-Befehle der Punktzahl der Polygone. In den Parametern 2 bis 4 stehen im Normalfall die Komponenten des eben angesprochenen Vektors. Hier gilt beim FS5DesignerTool aber die Vereinbarung, daß der Vektor genau dann selbständig aus den Punktdaten berechnet wird, wenn alle drei Vektorkomponenten auf "0" stehen. In unserem Fall werden also die Vektoren aller Polygone automatisch berechnet. In den nachfolgenden Parametern sind die Indices der verwendeten Punkte aufgelistet; diese Methode kennen wir ja schon vom Verfahren zur Erzeugung von SURFACES (S.16).

In diesem Beispiel haben wir für alle Polygone eine einzelne Farbe verwendet, indem wir allen POLYGON-Befehlen einen COLOUR-Befehl vorangestellt haben. Natürlich ist es auch möglich, jeder Wand des Hauses eine andere Farbe zuzuordnen:

COLOUR 10	
POLYGON 4, 0,0,0, 0,4,7,3	' Fläche vorne
COLOUR 11	
POLYGON 4, 0,0,0, 3,7,6,2	' Fläche rechts
COLOUR 12	
POLYGON 4, 0,0,0, 2,6,5,1	' Fläche hinten
COLOUR 13	
POLYGON 4, 0,0,0, 0,1,5,4	' Fläche links
COLOUR 15	
POLYGON 4, 0,0,0, 4,5,6,7	' Fläche Dach

Realistischere Objekte können Sie erzeugen, indem Sie anstatt einfachen Farben Objekte mit Farbverläufen definieren. Dafür existiert der Befehl "SHADED_POLYGON", der im Zusammenspiel mit dem Spezialbefehl "SHADE_COLOUR" den gewünschten Effekt bewirkt. Die Syntax des "SHADED_POLYGON"-Befehl ist identisch zu der des normalen "POLYGON"-Befehls. Sie können also das obige Beispiel auch schreiben als:

SHADE_COLOUR	10		' Farbe hellbraun
SHADED_POLYGON	4,	0,0,0, 0,4,7,3	' Fläche vorne
SHADED_POLYGON	4,	0,0,0, 3,7,6,2	' Fläche rechts
SHADED_POLYGON	4,	0,0,0, 2,6,5,1	' Fläche hinten
SHADED_POLYGON	4,	0,0,0, 0,1,5,4	' Fläche links
SHADED_POLYGON	4,	0,0,0, 4,5,6,7	' Fläche Dach

Vergessen Sie nicht, die Vereinbarung der einzelnen Punkte voranzustellen.

Neben Farben ist es auch möglich, Polygone mit Mustern zu füllen. Beispiele dafür finden sich unter anderem in den Szenarien, die zum Lieferumfang des FS5 gehören. Viele Bodenflächen sind mit vordefinierten Bitmaps angefüllt; daneben existieren (z.B. in der Chicago-Szenarie) Gebäude mit speziellen Beschriftungen ("Prudential"-Gebäude).

Drei Möglichkeiten existieren, um mit Bitmaps bzw. Texturen gefüllte Objekte zu erzeugen: Die erste Methode besteht darin, einen speziellen Polygon-Befehl zu benutzen, nämlich `TEX_POLYGON`. Dieser wird genau wie die beiden verwandten Befehle `"POLYGON"` und `"SHADED_POLYGON"` benutzt, d.h. zunächst definieren Sie sich eine Menge von Punkten (mit `DEFINE_POINTS` oder `DEFINE_ROTATED_POINTS`) und rufen diese dann mit `TEX_POLYGON` auf. Dem Polygon-Befehl stellen Sie nun aber nicht mehr die Definition einer Zeichen-Farbe voran, sondern die Definition einer Datei, in der die Bitmap vereinbart ist. Sie wählen die gewünschte Bitmap an mit dem Befehl `"BITMAP"`.

Allgemein hat der `BITMAP`-Befehl folgende Syntax:

```
BITMAP "<Name der Füllmuster-Datei, mit Extension!>",
      <Delta-North>, <Delta-East>, <Skalierungsfaktor aus [0..32767]>,
      <Dummy1>, <Dummy2>
```

In den Parametern 2 und 3 können Sie eine x- und y-Verschiebung angeben, mit der die `BITMAP` aufgerufen werden soll. Die Bitmap-Datei ist eine Matrix aus 256×256 Punkten, von denen jeder eine von 256 möglichen Farben annehmen kann; unseres Wissens läßt der FS5 auch keine anderen Texturen zu. Wenn Sie also in Parameter 2 oder 3 einen Wert verschieden von 0 angeben, erhalten Sie ein anderes Ergebnis.

Weit deutlicher ist dagegen die Wirkung des Skalierungsfaktors. Hier können Sie angeben, mit welcher Vergrößerung die Bitmap dargestellt werden soll. Kleine Werte bedeuten starke Vergrößerung, große Werte stehen für starke Verkleinerung der Bitmap.

Die Parameter 4 und 5 stellen weitere Einstellgrößen des `BITMAP`-Befehls dar; ihre genaue Funktion ist aber noch nicht klar. Verwenden Sie nach Möglichkeit in diesen Parametern immer den Wert "0", dann sollte es keine Schwierigkeiten geben.

Wenn wir also ein Polygon etwa mit der Bitmap namens "soldier.r8" füllen wollen, so schreiben wir:

```
BITMAP "soldier.r8",0,0,0,0,0 'lade Bitmap-Datei "soldier.r8" hinzu
TEX_POLYGON 4, 0,0,0, 0,4,7,3 ' Fläche vorne
TEX_POLYGON 4, 0,0,0, 3,7,6,2 ' Fläche rechts
TEX_POLYGON 4, 0,0,0, 2,6,5,1 ' Fläche hinten
TEX_POLYGON 4, 0,0,0, 0,1,5,4 ' Fläche links
TEX_POLYGON 4, 0,0,0, 4,5,6,7 ' Fläche Dach
```

Der FS5 sucht "seine" Bitmaps immer im Directory "texture", d.h. wenn Ihr FS5 etwa im Verzeichnis "C:\FS5" steht, wird er eine Bitmap-Datei immer im Verzeichnis "C:\FS5\TEXTURE" suchen. Achten Sie also immer darauf, daß Ihre Bitmap-Dateien im texture-Directory des FS5 stehen, bevor Sie sie benutzen. Der FS5 reagiert sonst mit einem Programmabbruch.

Die zweite Methode zur Darstellung von Bitmaps bezieht sich nur auf Bodenpolygone, die in einer Befehlsfolge `"BEGIN_SURFACE...END_SURFACE"` gezeichnet werden. Sie haben auch hier die Wahl, entweder eine Farbe vor die `"BEGIN_SURFACE...END_SURFACE"`-Sequenz zu schreiben, oder eine Bitmap anzuwählen per `"BITMAP"`. Verwenden Sie hier einfach denselben `BITMAP`-Befehl, den Sie auch vor einem `"TEX_POLYGON"`-Befehl verwenden würden. Wir nehmen uns noch einmal das bekannte Bodenpolygon aus dem Abschnitt über Straßen und Linien vor:

```
DEFINE_POINTS 4,0,0,0,0,1000,0,0,1000,1000,0,0,1000,0
BITMAP "soldier.r8",0,0,0,0,0 'lade Bitmap-Datei "soldier.r8" hinzu
BEGIN_SURFACE
MOVE_TO_POINT 0
LINE_TO_POINT 1
LINE_TO_POINT 2
LINE_TO_POINT 3
```



```

LINE_TO_POINT 0
END_SURFACE

```

Sie erhalten ein (ziemlich buntes) Bodenpolygon, in dem das Muster "soldier.r8" mehrere Male neben- und übereinander auftaucht.

Abschließend wollen wir die flexibelste und leistungsfähigste Variante zum Zeichnen von Bitmaps kennenlernen; es ist der Befehl "USE_BITMAP". Er besitzt weitere Parameter zum Einstellen der Darstellung der Bitmaps und eignet sich besser als die oben betrachtete Methode mit "TEX_POLYGON" dazu, Texturen bzw. Bitmaps auch an Gebäuden anzubringen, d.h. an Nicht-Boden-Polygonen. Darüber hinaus sind Staucheffekte möglich, wie sie beim Stadion in der FS5-Welt zu sehen sind ("soldier.r8"-Bitmap). USE_BITMAP weist gegenüber den bisher kennengelernten Polygon-Befehlen eine andere Syntax auf:

```

USE_BITMAP <Anzahl der Punkte des eingrenzenden Polygons>,
          <Latitude Vektor>, <Longitude Vektor>, <Altitude Vektor>,
          <Distanzwert in m aus [0..2^31]>,
          <Index des ersten Punktes>,
          <Bitmap-x des ersten Punkt 1>, <Bitmap-y des ersten Punktes>,
          ...
          <Index des letzten Punktes>,
          <Bitmap-x des letzten Punktes>, <Bitmap-y des letzten Punktes>

```

Bei Verwendung dieses Befehls brauchen Sie den Befehl POLYGON nicht mehr zu benutzen, "USE_BITMAP" erzeugt selber das Polygon. Was Sie allerdings benötigen, sind die Punkte aus DEFINE_POINTS. Diese werden wieder über ihre Nummern "adressiert". Daneben benötigen wir, wie bei den Polygon-Befehlen einen Vektor, der senkrecht auf dem Polygon steht und davon weg weist, bestehend aus Latitude-, Longitude- und Altitude-Anteil. Der Parameter "Distanzwert" sorgt - neben dem eben angesprochenen senkrechten Vektor - dafür, daß bei einem dreidimensionalen Objekt wie einem Haus auch nur die wirklich sichtbaren Seiten dargestellt werden. Die Berechnung dieses Wertes ist ziemlich komplex und soll hier nicht weiter erläutert werden.

Beide Parameterwerte, den senkrechten Vektor und den Distanzwert brauchen Sie nicht selbst zu bestimmen; das FS5DesignerTool nimmt Ihnen diese Arbeit ab, wenn Sie sowohl alle Komponenten des senkrechten Vektors als auch den Distanzwert einfach auf "0" setzen. Das FS5DesignerTool berechnet genau dann die korrekten Einstellungen dieser Parameter, wenn Sie zusätzlich dafür sorgen, daß die Punkte des Polygons im Uhrzeigersinn aufgerufen werden. Ob Sie die Punkte im Uhrzeigersinn aufrufen, legen Sie eindeutig fest, indem Sie von "außen" auf das Polygon schauen. Betrachten Sie die Punkte einfach in der Reihenfolge, wie sie in "USE_BITMAP" verwendet werden.

Wir wollen uns als ein kleines Beispiel ein aufrechtes "Fenster" ansehen, das mit dem "Prudential"-Schriftzug versehen ist.

```

DEFINE_POINTS 4,0,
              0,0,0,
              0,2000,0,
              0,2000,1000,
              0,0,2000
BITMAP "prudntl.r8",0,0,0,0,0
USE_BITMAP 4,0,0,0,0,
           0,0,200,
           1,200,100,
           2,200,0,
           3,0,0

```

Zu jedem Punkt geben Sie hier nicht nur seine Nummer an, sondern daneben auch zwei Koordinaten im Bereich [0..255]. Wie bereits oben erwähnt, besteht jede gültige Bitmap aus einer 256 x 256- Matrix. Wir können nun zu jedem Polygonpunkt angeben, welchem Punkt auf der 256 x 256-Matrix der Bitmap er entsprechen soll. Den Rest (Stauen / Drehen) erledigt dann der FS5 für uns. Versuchen Sie einmal andere x- / y-Werte bei den vier Punkten (etwa: Austauschen von Wertepaaren zwischen zwei Punkten), um diesen Effekt auch selber anschaulich nachvollziehen zu können.

Nach diesem Exkurs zu Texturen wollen wir uns noch einmal den Farben zuwenden. Wir haben oben ja schon gesehen, daß sich mit dem "SHADED_POLYGON" echte Farbverläufe erzeugen lassen. Manchmal kann es aber schon reichen, daß ein Polygon, etwa eine Seite eines Gebäudes, nur eine einzige Farbe erhält, die sich in Abhängigkeit von der Sonneneinstrahlung ändert. Bei der vorhandenen FS5-Palette ist dies bei einigen Farben der Fall (bei anderen dagegen nicht, z.B. wird Nr.15 = leuchtendrot nicht angepaßt).

An einer Farbe, die je nach Sonneneinstrahlung angepaßt wird, soll nun gezeigt werden, wie wirkungsvoll diese Eigenschaft benutzt werden kann. In diesem Beispiel wird zusätzlich noch ein neuer Befehl verwendet, "CONCAVE_POLY". Dieser soll dem FS5 anzeigen, daß das nachfolgende zu zeichnende Polygon konkav ist, weil solche Polygone im allgemeinen schwieriger zu zeichnen sind und vom FS5 weitere Sichtbarkeitsprobleme gelöst werden müssen. Hier das Beispiel:

```

DEFINE_POINTS 6,0,
    0,0,0,
    0,200,0,
    220,280,0,
    300,40,0,
    170,-80,0,
    150,30,300
' 6 Punkte
' Bodenzpunkt 1
' Bodenzpunkt 2
' Bodenzpunkt 3
' Bodenzpunkt 4
' Bodenzpunkt 5
' Spitze

GROUND_COLOUR 11
CONCAVE_POLY
POLYGON 3,0,0,0,1,0,5
POLYGON 3,0,0,0,2,1,5
POLYGON 3,0,0,0,3,2,5
POLYGON 3,0,0,0,4,3,5
POLYGON 3,0,0,0,0,4,5

```

Was könnte das für ein Objekt sein?

Zum Abschluß dieses Abschnitts über Polygone beim FS5DesignerTool wollen wir noch kurz einen abgewandelten Polygonbefehl ansehen, der sich wohl am ehesten mit den "POLYGON"-Befehlen vergleichen läßt. Er heißt schlicht "POLY" und unterscheidet sich vom "POLYGON"-Befehl dadurch, daß er ein anderes Verfahren zur Lösung von Sichtbarkeitsproblemen verwendet : Um zu entscheiden, wann eine Seite eines Objekts (etwa eines Hauses) dargestellt wird, verwendet "POLY" nicht wie "POLYGON" einen einzelnen Distanzwert, sondern gleich einen Distanz-Vektor. Seine Syntax lautet:

```

POLY <Anzahl der Punkte>,
    <Delta Latitude>, <Delta Longitude>, <Delta Altitude>,
    <Latitude Vektor>, <Longitude Vektor>, <Altitude Vektor>,
    <Indices der Punkte, getrennt durch Kommata>

```

Der Distanzvektor steht in den Parametern "Delta Latitude", "Delta Longitude" und "Delta Altitude". Die Berechnung dieses Vektors kann das FS5DesignerTool für Sie übernehmen; setzen Sie dazu einfach alle drei Komponenten des Vektors auf "0", wie schon von den senkrechten Vektoren der "POLYGON"-Befehle bekannt. Dieser senkrechte Vektor findet sich auch bei "POLY" wieder; und natürlich kann er auch wieder automatisch berechnet werden, indem alle Komponenten des senkrechten Vektors auf "0" gesetzt werden. Vergessen Sie auch hier nicht, die Punkte im Uhrzeigersinn aufzurufen. (Wenn ein solches Polygon mal nicht vom FS5 angezeigt wird, haben Sie die Punkte vermutlich in der falschen Reihenfolge aufgerufen. Kehren Sie dann einfach die Punktfolgenfolge im "POLY"-Befehl um.)

Unser schon bekanntes Beispiel läßt sich auch schreiben als:

```

DEFINE_POINTS 8,0,
    0, 0, 0,
    30, 0, 0,
    30, 40, 0,
    0, 40, 0,
    0, 0, 70,
    30, 0, 70,
    30, 40, 70,
    0, 40, 70
'Punkt links unten vorne
'Punkt links unten hinten
'Punkt rechts unten hinten
'Punkt rechts unten vorne
'Punkt links oben vorne
'Punkt links oben hinten
'Punkt rechts oben hinten
'Punkt rechts oben vorne

COLOUR 10
'Farbe hellbraun
POLY 4, 0,0,0, 0,0,0, 0,4,7,3,
'Fläche vorne
POLY 4, 0,0,0, 0,0,0, 3,7,6,2,
'Fläche rechts
POLY 4, 0,0,0, 0,0,0, 2,6,5,1
'Fläche hinten
POLY 4, 0,0,0, 0,0,0, 0,1,5,4
'Fläche links
POLY 4, 0,0,0, 0,0,0, 4,5,6,7
'Fläche Dach

```

4.5 Bedingungen und Unterprogramme

Der FS5 bietet bei der Verwendung von einfachen Operationen einige strukturierende Hilfen an; diese dienen mehreren Zwecken.

- mehrfache Verwendung desselben Codes durch Verwendung von Unterprogrammen
- Strukturierung des Programmablaufs durch Verzweigungen / Bedingungen

- Abfrage von internen FS5-Variablen (z.B. Tageszeit, Flughöhe)
- Beschleunigung der Abarbeitung der Szeneriedateien

Wie läßt sich die Abarbeitung von Dateien beschleunigen? Der FS5 verbringt sehr viel Zeit damit, immer den "richtigen" Ausschnitt des aktuellen Szeneriefles auszuwählen, um ihn darzustellen. Das bedeutet, der FS5 muß immer weite Teile der Szeneriedaten überspringen, um so möglichst schnell den richtigen Ausschnitt zu finden. Durch die Vereinbarung von Unterprogrammen ermöglichen wir dem FS5 ein schnelleres "Überlesen" von Codesequenzen. Die Beschleunigung der Abarbeitung beruht also darauf, daß ungünstige Codesequenzen schneller überlesen werden.

Wenn wir also dafür sorgen wollen, daß eine Codesequenz ohne Verzögerungen abgearbeitet wird, obwohl sie evtl. sehr komplex ist (z.B. große Polygone), so können wir diese als "Unterprogramm" vereinbaren. Dafür bietet das FS5DesignerTool den Komplexbefehl "AUTO_CALL". Ein "AUTO_CALL", besagt, daß die nachfolgende Codesequenz ein Unterprogramm ist. Das Unterprogramm ist - wie alle anderen Unterprogramme auch - mit einem "RETURN" abzuschließen.

Wir könnten unsere Sequenz zum **Zeichnen eines Berges** also auch wie folgt schreiben:

```
AREA_TEST (N51,36,55), (E8,37,2),255
REF_POINT (N51,36,55), (E8,37,2),0,1,1
AUTO_CALL                                     ' Beginn des Unterprogramms
DEFINE_POINTS 6,0,0,0,0,0,200,0,220,280,0,
              300,40,0,170,-80,0,50,30,300
GROUND_COLOUR 11
CONCAVE_POLY
POLYGON 3,0,0,0,1,0,5
POLYGON 3,0,0,0,2,1,5
POLYGON 3,0,0,0,3,2,5
POLYGON 3,0,0,0,4,3,5
POLYGON 3,0,0,0,4,5
RETURN                                     ' Ende des Unterprogramms
```

In diesem Beispiel steht die Definition des Referenzpunktes im Unterprogramm selbst. Einzig den AREA_TEST dürfen wir nicht im Unterprogramm vereinbaren, diese einfache Operation nimmt gegenüber den übrigen einfachen Operationen eine gewisse Sonderstellung ein.

Die Verwendung des Unterprogramms hat in unserem Beispiel aber keine Auswirkung auf das Aussehen des gezeichneten Berges. Für den Moment genügt es einfach zu wissen, daß der "AUTO_CALL" immer dann eingesetzt werden kann, wenn große zusammenhängende Codesequenzen schnell ausgeführt werden sollen.

Und damit wollen wir uns auch gleich den "echten" Unterprogrammen zuwenden; darunter verstehen wir solche Codesequenzen, die nicht unmittelbar auf die Aufrufstelle im Code folgen. Ein Beispiel dazu wäre etwa:

```
GROUND_COLOUR 15      ROTATED_CALL 0,0,0,"Straße"
GROUND_COLOUR 20      ROTATED_CALL 0,0,90,"Straße"
GROUND_COLOUR 10      ROTATED_CALL 0,0,180,"Straße"
GROUND_COLOUR 5       ROTATED_CALL 0,0,270,"Straße"
JUMP "Fertig"
LABEL "Straße"        ' Unterprogramm zeichnet Straße
START_SUB
ROAD_MOVE_TO1 0,0,0,30      ROAD_LINE_TO1 1000,0,0
RETURN
LABEL "Fertig"
```

In diesem Beispiel werden sowohl der komplexe Unterprogrammsprung "ROTATED_CALL" als auch der einfache Sprung "JUMP" verwendet. Das Sprungziel ist immer angegeben als ein String. Sprünge sind beim FS5DesignerTool immer Vorwärtssprünge. Das Sprungziel wird im späteren Code durch Setzen eines "LABEL"s gekennzeichnet. Der FS5 springt also beim Aufruf 'ROTATED_CALL 0,0,0,"Straße"' an die nächste Stelle im Code, die durch 'LABEL "Straße"' gekennzeichnet ist. Steht im weiteren Verlauf des Codes erneut das 'LABEL "Straße"', so springt der FS5 trotzdem nicht dorthin; es wird immer die nächste "LABEL"-Vereinbarung angesprungen.

Achten Sie bei der Verwendung von Labels als Sprungzielen bitte immer darauf, daß das Label an Sprungstelle und Sprungzielstelle identisch sein muß; insbesondere muß die Groß- / Kleinschreibung identisch übernommen werden.

Am Beginn des Unterprogramms steht der Befehl "START_SUB", ohne Parameter. In manchen Fällen kann es notwendig sein, am Beginn eines längeren Unterprogramms diesen Befehl einzufügen. Treten im Zusammenhang mit Unterprogrammen beim FS5 also einmal Probleme auf, so fügen Sie einfach diesen Befehl ein.

Wir haben in diesem Beispiel einen komplexen Unterprogrammaufruf kennengelernt, "ROTATED_CALL". Dieser sorgt dafür, daß alle primitiven Operationen des aufgerufenen Unterprogramms nicht in der angegebenen Orientierung ausgeführt werden, sondern mit einer Drehung, die sich in x-, y- und z-Richtung angeben läßt.

Die ersten drei Parameter des "ROTATED_CALL"-Befehls beschreiben die x-, y- und z-Anteile der Drehung. In diesem Beispiel haben wir nur um die z-Achse drehen lassen, und dies wird auch in den meisten Fällen ausreichend sein.

Natürlich kennt der FS5 auch einen einfachen Unterprogrammaufruf, ohne Drehung. Dieser heißt schlicht "CALL". Damit lassen sich - ähnlich dem "AUTO_CALL"-Befehl - Ausschnitte von Szenarien so vereinbaren, daß der FS5 sie besonders schnell "lesen" kann. Wir geben dazu ein kleines Beispiel.

```
AREA_TEST (N51,36,55), (E8,37,2),255
CALL "Haus1"           ' rufe Unterprogramme zum Zeichnen der Häuser auf
CALL "Haus2"
CALL "Haus3"
JUMP "fertig"          ' springe hinter die Unterprogramme

LABEL "Haus1"
  START_SUB
  REF_POINT (N51,36,55), (E8,37,2),0,1,1,"Ende1"
  BUILDING 0,0,0,30,30,60,5,511
  LABEL "Ende1"
RETURN
LABEL "Haus2"
  START_SUB
  REF_POINT (N51,36,45), (E8,37,2),0,1,1,"Ende2"
  BUILDING 0,0,0,30,30,60,5,511
  LABEL "Ende2"
RETURN
LABEL "Haus3"
  START_SUB
  REF_POINT (N51,36,55), (E8,37,8),0,1,1,"Ende3"
  BUILDING 0,0,0,30,30,60,5,511
  LABEL "Ende3"
RETURN
LABEL "fertig"
```

In diesem Beispiel kommt auch der "JUMP"-Befehl zur Verwendung; diesen werden wir gleich noch genauer betrachten. Ähnliches gilt für den Befehl "BUILDING" zum Zeichnen von Gebäuden verschiedenster Art.

An unserem Beispiel können wir sehen, daß für einen Unterprogrammaufruf "CALL" kein Referenzpunkt gesetzt worden sein muß; dies geschieht vielmehr in den Unterprogrammen selbst. Jedes Gebäude erhält seinen eigenen Bezugspunkt. Diese Methode mag auf den ersten Blick etwas unhandlich erscheinen -- schließlich hätte man ja die Befehle zum Zeichnen der Gebäude ja auch einfach nacheinander schreiben können, mit nur einem Bezugspunkt.

Wir kommen damit auf ein grundsätzliches Problem von Sichtbarkeiten im FS5-Szenariebau zu sprechen, das immer dann entsteht, wenn mehrere Objekte sich gegenseitig überdecken können. Bei den drei Hochhäusern in unserem Beispiel ist das immer dann der Fall, wenn man im FS5 eine Perspektive wählt, in der mindestens zwei der Häuser knapp hintereinander angeordnet sind. In unserer Szenerie entstehen keine Sichtbarkeitsprobleme; das aus der aktuellen Perspektive vorderste Haus wird auch als "vorne" dargestellt, wodurch das hintere Haus teilweise oder ganz verdeckt ist. Der FS5 löst dieses Sichtbarkeitsproblem quasi automatisch, wenn wir jedes Gebäude einzeln vereinbaren.

Vielleicht ist Ihnen auch aufgefallen, daß die "REF_POINT"-Befehle in den Unterprogrammen einen zusätzlichen Parameter in Form eines Strings aufweisen. Dieser Parameter ist ein optionaler Parameter, das heißt, Sie können ihn normalerweise auch weglassen. In ihm teilen Sie dem FS5 mit, wohin er springen soll, wenn er den zu diesem Sichtbarkeitsbereich gehörenden Code abgearbeitet hat. Im Normalfall ist dies der nachfolgende

"AREA_TEST", und das ist auch der Wert, den das FS5DesignerTool automatisch erzeugt, wenn wir den optionalen Parameter weglassen.

Wollen wir allerdings auch in Unterprogrammen einen Referenzpunkt setzen, so müssen wir dem FS5 im "REF_POINT"-Befehl mitteilen, wo das Unterprogramm endet und der FS5 wieder die Kontrolle übernimmt. Ein Unterprogramm endet immer mit dem Rücksprung zur Aufrufadresse, also mit einem "RETURN". Genau dies ist auch die Stelle, die wir im "REF_POINT"-Befehl angeben müssen. Wohlgedermt, diese Maßnahme ist nur dann erforderlich, wenn wir einen "REF_POINT"-Befehl innerhalb eines Unterprogramms verwenden wollen.

Einen besonders schönen Effekt können Sie übrigens erzielen, wenn Sie ein solches Unterprogramm zweimal aufrufen, nämlich einmal mit "CALL" und einmal mit einem "SHADOW_CALL". Der "SHADOW_CALL" sorgt dann dafür, daß nicht die Objekte des Unterprogramms selbst, sondern ein Schatten dafür dargestellt wird. Wir erweitern unser obiges Beispiel mal um diesen Befehl:

```
AREA_TEST (N51,36,55), (E8,37,2),255
CALL "Haus1"          SHADOW_CALL "Haus1a"
CALL "Haus2"          SHADOW_CALL "Haus2a"
CALL "Haus3"          SHADOW_CALL "Haus3a"
JUMP "fertig"          ' springe hinter die Unterprogramme
LABEL "Haus1"         START_SUB
    LABEL "Haus1a"     REF_POINT (N51,36,55), (E8,37,2), 0,1,1, "Ende1"
    BUILDING 0,0,0,30,30,60,5,511
    LABEL "Ende1"      RETURN
RETURN
LABEL "Haus2"         START_SUB
    LABEL "Haus 2a"    REF_POINT (N51,36,45), (E8,37,2), 0,1,1, "Ende2"
    BUILDING 0,0,0,30,30,60,5,511
    LABEL "Ende2"      RETURN
RETURN
LABEL "Haus3"         START_SUB
    LABEL "Haus3a"     REF_POINT (N51,36,55), (E8,37,8), 0,1,1, "Ende3"
    BUILDING 0,0,0,30,30,60,5,511
    LABEL "Ende3"      RETURN
RETURN
LABEL "fertig"
```

Sie sehen vielleicht schon, daß das Ziel der "SHADOW_CALL"s immer hinter den "START_SUB"-Befehlen liegt. Zum Zeitpunkt der Erstellung dieses Handbuchs ist dem Autor aber noch nicht klar, ob das immer der Fall ist.

So, damit ist die Familie der "CALL"-Befehle komplett, wir können uns nun den Sprungbefehlen widmen. Den einfachen Sprung, "JUMP", haben wir ja bereits im Beispiel zu "CALL" kennengelernt. Aber der FS5 bietet noch mehrere bedingte Befehle. Bei diesen werden entweder die Augenblicksposition abgefragt, oder es werden interne Variablen des FS5 getestet.

Der erste Befehl, den wir betrachten wollen, heißt "JUMP_ON_VAR_NOT". Damit können testen, ob eine bestimmte FS5-Variablen einen bestimmten Wert aufweist oder nicht. Sind die Werte verschieden, so springt der FS5 an das angegebene Sprungziel.

Hier zunächst ein Beispiel:

```
DEFINE_POINTS 2,0,0,0,205,0,0,250
LINE_COLOUR 4
MOVE_TO_POINT 0
LINE_TO_POINT 1
JUMP_ON_VAR_NOT 652,-2,"tagsüber"
LINE_COLOUR 20
DOTTED_LINE 0,0,205,0,0,250,5
LABEL "tagsüber"
BUILDING 0,0,0,20,37,50,7,31
```

Diese Codesequenz erzeugt ein hohes Gebäude, mit einer langen Antenne auf dem Dach. Den neuen Befehl "BUILDING" zum Erzeugen von Gebäuden werden wir später noch ausführlicher erläutern. Jetzt wollen wir mal den "JUMP_ON_VAR_NOT" genauer unter die Lupe nehmen.

Wenn Sie sich die erzeugte Szenerie ansehen, werden Sie feststellen, daß da nur ein Gebäude mit einer Linie auf dem Dach angezeigt wird - jedenfalls dann, wenn Sie sich die Einstellung bei Tageslicht ansehen. Verändern Sie doch einmal die Zeiteinstellung "time of day" auf "night". Das Gebäude ist kaum mehr zu erkennen, außer ein paar erleuchteten Fenstern. Jetzt tritt aber erstmals der "DOTTED_LINE"-Befehl in Erscheinung, in Form von fünf roten Punkten, die die Antenne auf dem Dach markieren.

Das muß mit dem "JUMP_ON_VAR_NOT"-Befehl zusammenhängen, denn wenn Sie ihn entfernen oder in Kommentar setzen, werden die roten Punkte auch während der hellen Stunden dargestellt. Was passiert in diesem Befehl?

Im ersten Parameter befindet sich die Nummer der internen Variable des FS5, die auf den Wert "-2" abgefragt wird. Ist der Wert der Variablen verschieden von "-2", so springt der FS5 an das angegebene Sprungziel, überspringt also das Zeichnen der "DOTTED_LINE". Das bedeutet: Ist der Wert der internen Variablen mit der Nummer 652 verschieden von "-2", so ist es hell.

Einen weiteren bedingten Befehl stellt JUMP_ON_VAR_RANGE dar. Mit diesem können Sie testen, ob eine interne FS5-Variable innerhalb eines bestimmten Wertebereichs liegt. Ist dies nicht der Fall, springt der FS5 an die durch das Sprungziel angegebene Stelle. Hier wieder ein praktisches Beispiel:

```
JUMP_ON_VAR_RANGE 898,0,1000, "zu hoch"      ' Höhentest
LINE_COLOUR 15                                ' zeichne Linie nur unter 4000 Fuß
DOTTED_LINE 0,0,0,250,0,0,50
LABEL "zu hoch"
```

Wenn Sie dieses Beispiel einmal ausprobieren, werden Sie feststellen, daß die gestrichelte Linie ("DOTTED_LINE") nur dann dargestellt wird, wenn Sie eine bestimmte Flughöhe unterschreiten. Im "JUMP_ON_VAR_RANGE"-Befehl wird hierfür die interne Variable mit der Nummer 898 abgefragt. In ihr ist die Flughöhe in Metern gespeichert. Genaugenommen wird die Linie nur dargestellt, wenn Ihre Flughöhe im Intervall 0m...1000m liegt. Sie finden diese Intervallgrenzen in den Parametern 2 und 3 wieder.

Diese Methode können Sie immer dann nutzen, wenn Sie das langwierige Zeichnen von komplexen Gebäuden oder Bodenflächen ab einer bestimmten Flughöhe unterdrücken wollen.

Als letzten bedingten Sprung in diesem Abschnitt wollen wir den Befehl "JUMP_ON_VECTOR" ansprechen. Hiermit können Sie feststellen, wie weit Ihre Augenblicksposition von demjenigen Referenzpunkt entfernt ist, der vor dem "JUMP_ON_VECTOR"-Befehl definiert wurde. Allgemein lautet die Syntax von "JUMP_ON_VECTOR" wie folgt:

```
JUMP_ON_VECTOR <Vektor Latitude in m>, <Vektor Longitude in m>,
               <Vektor Altitude in m>, <Wert 1>, <Wert 2>,
               "<Name der Sprungmarke>"
```

Die Bedeutung der Wechselwirkung der Parameter "Wert1" und "Wert2" ist noch nicht ganz klar. In den meisten Fällen wird es genügen, "Wert1" auf die zu testende Distanz zu setzen und "Wert2" auf "0".

4.6 Weitere einfache Operationen

In diesem letzten Abschnitt wollen wir noch einige Einzeloperationen kennenlernen, die sich keiner übergeordneten Befehlsgruppe zuordnen lassen. Wir gehen nach dem Alphabet vor.

Der erste Befehl, "BLINK", setzt einen gelben Punkt am vorher definierten Referenzpunkt. Dieser blinkende Punkt kann zu Markierungszwecken gesetzt werden. Der Befehl hat keinen Parameter.

Der zweite Befehl, "BUILDING", ist schon etwas komplexer. Sie können damit in einem Befehl ganze Gebäude erzeugen, die vom FS5 auch als Hindernis erkannt werden (im Gegensatz zu den Gebäuden, die als Polygone definiert wurden). Ein solches Gebäude wird beschrieben durch seine Position, seine Länge und Breite und die Anzahl der Stockwerke.

Daneben können verschiedene "Fenster-Muster" angegeben werden, die wie folgt codiert sind:

Wert	Bedeutung
0	Beige mit vertikalen Streifen
1	Weiß mit vertikalen Streifen
3	Dunkelgrau mit vertikalen Streifen
4	Weiß mit dunklen vertikalen Streifen
5	Weiß mit horizontalen Streifen
6	Beige mit horizontalen Streifen
7	Hellgrau mit horizontalen Streifen

Wenn dem "Fenster"-Parameter noch "8" hinzuaddiert werden, erhalten wir ein Gebäude mit Schrägdach. Bei Addition von "24" erhalten wir ein Gebäude mit 8 Seitenwänden.

Schließlich gibt es einen Parameter zum Beschreiben der Gebäudestruktur, insbesondere der Anzahl der begrenzenden Wände. Dazu wollen wir uns einmal ein kleines Beispiel ansehen:

```
BUILDING 0,0,0, ' Position
          20,20, ' Länge / Breite
          10,    ' Anzahl Stockwerke
          4,     ' Fenstermuster
          31     ' binär 0001 1111 --> die unteren 5 Bits sind auf
                  ' "1" -> 5 Wände (4 Seiten, 1 Dach)
```

Versuchen Sie mal dasselbe Gebäude mit Fenstermuster "4+24" und Wandparameter "-1". Soviel auch zum "BUILDING"-Befehl; experimentieren Sie ruhig ein wenig weiter.

Eine Hilfe zum Austesten von bisher unbekannten Befehlen stellt der "INLINE"-Befehl dar. Hiermit haben Sie die Möglichkeit, eine einfache Operation durch explizite Angabe der sie bildenden Bytefolge einzufügen. Ein Beispiel hierfür ist etwa:

```
INLINE 82,0, ' Befehlscode "GROUND_COLOUR"
       15,   ' Farbnummer 15
       240   ' "MAGIC NUMBER"
```

Diese Bytesequenz entspricht dem Befehl 'GROUND_COLOUR 15'. Seien Sie sich aber bewußt, daß die Eingabe von undefinierten Codefolgen zum Absturz bzw. nicht vorhersagbarem Verhalten des FS5 führen kann. Verwenden Sie diesen Befehl also nur, wenn Sie ganz sicher sind, daß Ihr Code keinen Schaden anrichten kann.

Wir haben gerade eben erst den Befehl "BLINK" kennengelernt, zum Setzen eines Punktes. Eine flexiblere Variante stellt der Befehl "POINT" dar, der ebenfalls einen blinkenden Punkt erzeugt. Sie können darin aber auch eine relative Verschiebung angeben (das heißt, der Punkt muß nicht - wie bei BLINK - mit dem Referenzpunkt identisch sein). Außerdem dürfen Sie sich hier zwei Farben aussuchen, in denen der Punkt nacheinander aufleuchten soll. Wollen wir also einen Punkt mit 10 m Verschiebung in Latitude-Richtung, 20m in Longitude-Richtung und 5 m in Altitude-Richtung, der sowohl gelb als auch rot blinkt, so schreiben wir einfach

```
POINT 10,20,5, ' relative Verschiebung
      20,15    ' Farbwerte 1 und 2
```

Zum Setzen von Sichtbarkeitsbereichen haben wir bisher nur den "AREA_TEST" verwendet, für den es immer notwendig war, eine absolute Koordinate anzugeben. Nun kann es beim Entwerfen von langen Straßenzügen - also beispielsweise einer Autobahn, die sich über mehrere hundert Kilometer erstrecken kann -, günstig sein, wenn die neuen Sichtbarkeitsbereiche nicht immer wieder absolut angegeben werden, sondern in einer "relativen" Darstellung, bezogen auf den zuletzt erfolgten "AREA_TEST". Ein ähnliches Verfahren kennen Sie ja z.B. schon vom Befehl "ROAD_DRAW1", bei dem mit Verschiebungsdistanzen und Richtungswinkeln gearbeitet wird. Eine ähnliche Möglichkeit stellt der Befehl "REL_AREA_TEST1" zur Verfügung: Haben Sie erst einmal einen Sichtbarkeitsbereich über seine absoluten Koordinaten vereinbart, können Sie von dann an jeden nachfolgenden Sichtbarkeitsbereich definieren, indem Sie ihn auf den zuletzt gemachten "AREA_TEST" beziehen. Ein Beispiel wäre also:

```
AREA_TEST (N51,36,55), (E8,37,2), 255 ' erster Sichtbarkeitsbereich
... ' Objekte dafür
REL_AREA_TEST1 10000,90,255 ' neuer Sichtbarkeitsbereich mit
                             ' 10000 m Distanz in Richtung 90°
                             ' von (N51,36,55), (E8,37,2)
```

Alternativ zum "REL_AREA_TEST1" steht noch der "REL_AREA_TEST2" zur Verfügung; dort können Sie auch eine Verschiebung in Metern angeben, aufgelöst nach Latitude-Richtung und Longitude-Richtung. Obiges Beispiel könnte also auch geschrieben werden als:

```
AREA_TEST (N51,36,55), (E8,37,2), 255 ' erster Sichtbarkeitsbereich
... ' Objekte dafür
REL_AREA_TEST1 0,10000,255 ' neuer Sichtbarkeitsbereich mit 0 m
                             ' Distanz in Latitude-Richtung und
                             ' 10000 m Distanz in Longitude-
                             ' Richtung von (N51,36,55), (E8,37,2)
```

Der Mechanismus der relativen "AREA_TEST"s erweist sich aber nur dann als sinnvoll, wenn auch die Referenzpunkte mithilfe dieser Methode vereinbart werden können. Dafür stehen die analogen Befehle "REL_REF_POINT1" und "REL_REF_POINT2" zur Verfügung. Wir erweitern dazu unsere obigen Beispiele.

Beispiel 1:

```

AREA_TEST (N51,36,55), (E8,37,2),255      ' erster Sichtbarkeitsbereich
REF_POINT (N51,36,55), (E8,37,2),228,1,1  ' Referenzpunkt
...                                         ' Objekte dafür
REL_AREA_TEST1 10000,90,255               ' neuer Sichtbarkeitsbereich mit
                                           ' 10000 m Distanz in Richtung 90°
                                           ' von (N51,36,55), (E8,37,2)
REL_REF_POINT1 10000,90,228,1,1           ' neuer Referenzpunkt mit 10000 m
                                           ' Distanz in Richtung 90° von
                                           ' (N51,36,55), (E8,37,2) in Höhe 228m

```

Beispiel 2:

```

AREA_TEST (N51,36,55), (E8,37,2),255      ' erster Sichtbarkeitsbereich
REF_POINT (N51,36,55), (E8,37,2),228,1,1  ' Referenzpunkt
...                                         ' Objekte dafür
REL_AREA_TEST1 0,10000,255                ' neuer Sichtbarkeitsbereich mit 0 m
                                           ' Distanz in Latitude-Richtung und
                                           ' 10000 m Distanz in Longitude-
                                           ' Richtung von (N51,36,55), (E8,37,2)
REL_REF_POINT2 0,10000,0,1,1              ' neuer Referenzpunkt mit 0 m
                                           ' Distanz in Latitude-Richtung,
                                           ' 10000 m in Longitude-Richtung und!
                                           ' 0 m in Altitude-Richtung

```

Wir kommen nun zur letzten einfachen Operation, "SET_VAR". Wir haben ja bereits im Abschnitt über Bedingungen und Unterprogramme, wie sich bestimmte Variablen sinnvoll nutzen lassen. "SET_VAR" kann dazu benutzt werden, eine bestimmte interne Variable des FS5 zu verändern, wenn Sie sich an einer bestimmten Position befinden. Ein Beispiel hierfür wäre eine Einrichtung an einem Flughafen zum Auftanken und Warten des Flugzeugs.

Um ein Flugzeug zu tanken und zu warten, müßten Sie etwa

```
SET_VAR 648, 1
```

eingeben.

5. Landschaftsbau einmal einfach - Die Sectionen 1 bis 6

Bisher haben wir in unseren Beispielszenarien immer nur einen graublauen Ton als Bodenfarbe gesehen, der sich nie änderte. Wir wollen in diesem Kapitel untersuchen, wie sich mit dem FS5 Designer Tool realistisch wirkende Landschaften mit Feldern, Städten, Wäldern und Bergen erzeugen lassen.

Die Erzeugung von Gewässern sparen wir hier bewußt aus, weil wir der Ansicht sind, daß sich etwa Flüsse und Seen beim FS5 am einfachsten durch Angabe von Polygonen abbilden lassen. Der FS5 stellt hierfür zwar auch leistungsfähige Befehle zur Verfügung, die auch den "Landschafts"-Befehlen recht nahe kommen, für den normalen Benutzer aber zu komplex sind.

5.1 Die sechs Sectionen

Im Abschnitt über die verschiedenen Koordinatendarstellungen beim FS5 haben wir bereits erfahren, daß der FS5 auch sogenannte "Blockkoordinaten" verwendet. Eine Koordinate dieses Typs beschreibt nicht mehr nur einen ausgewählten Punkt auf der Erdoberfläche, sondern vielmehr eine quadratische Fläche; wir wollen diese Fläche einen "Block" nennen. Damit ist der Namensursprung dieses Koordinatentyps geklärt.

Wenn wir beim FS5 eine Landschaft definieren wollen, werden wir im Allgemeinen nicht nur einen bestimmten Punkt, sondern eine größere Fläche damit beschreiben wollen. Der FS5 unterstützt sechs verschieden große Flächen, also "Blöcke", mit denen Landschaft beschrieben werden kann. Die größte Blockgröße haben Blöcke vom Typ 1 -- hier ist ein Block etwa 1° 24' 32" lang bzw. breit. Ein Block vom Typ 2 hat nur noch die halbe Kantenlänge wie ein Block des Typs 1, also etwa 42' 16". Jeder nachfolgende Blocktyp hat wiederum die halbe Kantenlänge des Vorgängers. Das bedeutet, daß wir beim kleinsten Typ, Typ 6, eine Kantenlänge von nur noch zwei Minuten und 38.5 Sekunden erhalten. Eine Tabelle mit den genauen Kantenlängen ist im Kapitel "Blockkoordinate" abgedruckt.

Wir können unsere Landschaften also von sehr grob (Section 1) bis sehr detailliert (Section 6) beschreiben.

Nun ist es aber leider nicht so, daß wir eine absolute Koordinate unmittelbar in die entsprechende Blockkoordinate umrechnen können, denn es besteht kein linearer Zusammenhang zwischen diesen beiden Darstellungsformen. (Ein "linearer" Zusammenhang würde bestehen, wenn wir einen Latitude-Wert X° dadurch in den entsprechenden Blockkoordinatenwert umrechnen könnten, indem wir einfach X° durch die oben angegebene Blockkantenlänge in Grad dividieren könnten.) Das hängt mit der Koordinatenabbildung des FS5 zusammen.

Als kleine "Arbeitserleichterung" haben wir im Anhang für jede Section eine Umrechnungstabelle von Absolutkoordinaten in Blockkoordinaten abgedruckt.

5.2 Landschaftstypen

Als ersten Befehl zur Definition einer Landschaft wollen wir jetzt "TERRAIN" untersuchen. Beispielhaft vereinbaren wir dazu eine Feldlandschaft um den Flughafen Paderborn-Lippstadt, d.h. bei der Koordinate (N51,36,55), (E8,37,2). Wir wählen zunächst die gröbste Darstellung, also Section 1, weil sich so auch die größten "Blöcke" erzeugen lassen.

Die allgemeine Syntax des TERRAIN-Befehls lautet wie folgt:

```
TERRAIN <Latitude in der Größe der Blöcke>,  
        <Longitude in der Größe der Blöcke>, <Section aus [1..6]>,  
        <Altitude in m>, <Landschaftstyp aus {0,1,2,3}>,  
        <Selektor1>, <Selektor2>, <Selektor3>
```

Der Latitude-Wert (N51,36,55) entspricht laut Tabelle der Blockkoordinate 42, der Longitude-Wert (E8,37,2) der Blockkoordinate 6.

Neben der Latitude- und Longitude-Koordinate können wir für unseren Block auch festlegen, in welcher Höhe er sich befinden soll. Wir wählen den Wert 228 m. Mit dem Befehl "TERRAIN" können wir also den Boden, der bisher immer nur in Höhe 0 m lag, auf beliebige Höhen "anheben".

In den letzten vier Parametern vereinbaren wir schrittweise, welcher Landschaftstyp dargestellt werden soll. Zunächst stehen einmal vier Basistypen zur Auswahl: Flache Landschaft (Typ 0), Berge (Typ 1) und zwei See-Typen (Typ 3 und 4). Da wir weder Berge noch Seen wollen, wählen wir den Basistyp 0.

Nun können wir in Selektor 1 weiter differenzieren. Im Anhang befindet sich bei der Beschreibung des "TERRAIN"-Befehls eine Auflistung der erlaubten Werte für den Selektor 1, abhängig vom gewählten Basistyp. Wir wählen einmal die "grüne Landschaft mit Hecken", das ist Einstellung "4" in Selektor 1.

Die Parameter "Selektor 2" und "Selektor 3" können verwendet werden, um die bisherigen Einstellungen weiter ausdifferenzieren. Ihre Bedeutungen sind aber noch unklar. Für die meisten Fälle genügt es, hier als Wert die "0" einzusetzen. Damit lautet unsere Definition:

```
TERRAIN 42, 6, 1, 228, 0, 4, 0, 0
```

Zum Austesten dieser Einstellung versuchen Sie doch einmal z.B. in der Grobverstellung, mit dem Flugzeug auf den Boden abzusinken. Wenn alles richtig war, sollten Sie bei etwa 700 Fuß stehenbleiben.

Wenn Sie mögen, können Sie auch schon einmal andere Landschaftstypen versuchen. Versuchen Sie mal Einstellung "142" für Selektor 1. Dann erhalten Sie eine graue Betonlandschaft mit Hochhäusern.

5.3 Das Streifen-Konzept

Wenn Sie größere Szenarien bauen wollen, so kann es sehr mühsam sein, alle Landschaftsblöcke einzeln mit dem "TERRAIN"-Befehl einzugeben. Da kommt uns der FS5 einmal mehr entgegen, indem er eine kompaktere Definition von großflächigen Landschaften erlaubt. Die kompaktere Darstellung erreichen wir dadurch, daß wir Blöcke, die aneinander angrenzen, in einem einzelnen Befehl zusammenfassen.

Einzige Bedingung neben der Tatsache, daß die Blöcke unmittelbar aufeinander folgen, ist daß sie alle denselben Latitude-Wert in der Blockkoordinate besitzen. Auf diese Weise erreichen wir quasi Querstreifen von Landschaft, das heißt, Landschaftsstreifen in West-Ost-Richtung.

Die einzelnen Blöcke sind in ihrer Definition aber wieder genauso präzise zu definieren, wie bereits aus dem "TERRAIN"-Befehl bekannt. Das heißt, für jeden Block können wir wieder seine Höhe in m, den Basistyp und die drei Selektoren angeben. Auf diese Weise büßen wir durch Verwendung der Streifen nichts an Flexibilität ein.

Wir wollen einmal einen solchen Streifen definieren, indem wir das Beispiel aus dem letzten Kapitel vorne und hinten "verlängern":

```
STRIPE 42, 5, 9, 1,
228, 0, 4, 0, 0,
228, 0, 4, 0, 0,
228, 0, 4, 0, 0,
228, 0, 4, 0, 0,
228, 0, 4, 0, 0
```

In Parameter 1 legen wir die Latitude-Komponente fest. Dann folgen der Longitude-Startwert sowie der Longitude-Endwert. Das bedeutet, wir durchlaufen in diesem Beispiel die Blockkoordinaten (42,5), (42,6), (42,7), (42,8) und (42,9). Der Latitude-Wert bleibt konstant, während der Longitude-Wert quasi eine Schleife durchläuft.

Parameter 4 definiert die Section, das ist der Blocktyp. Typ 1 erzeugt die größten Blöcke. Dann folgen, für alle Longitude-Werte (von 5 bis einschließlich 9), die Spezifikationen der Landschaftsblöcke (Höhe, Basistyp und Selektoren). In unserem Beispiel haben wir nur einen Landschaftstyp für alle Blöcke verwendet; natürlich lassen sich aber auch unterschiedliche Landschaftstypen verwenden.

Wenn Sie beim Basistyp 0 andere Werte für "Selektor 1" verwenden wollen, als in der Tabelle im Anhang angegeben, also insbesondere die Einstellungen "13" bis "127" und "144" bis "255", so versucht der FS5 bei Anwahl dieses Landschaftstyps eine Texture-Datei namens

```
"seed"+(Sedezimalwert von "Selektor 1")+".r8"
```

einzulesen. Bei einem "Selektor 1"-Wert von 32 würde etwa die Datei "seed20.r8" geladen. Diese Datei muß dann im "texture"-Directory des FS5 stehen.

Standardmäßig sind diese aber nicht im texture-Directory enthalten, so daß Sie sich hier eigene Texturen definieren können, um neue Landschaftstypen zu kreieren.

5.4 Brauchen wir Grenzen?

Gemeint sind hier nicht politische oder geographische Grenzen, sondern wir wollen in diesem Abschnitt klären, welche Vorteile für uns entstehen, wenn wir dem FS5 mitteilen, daß unsere Szenerie nur in einem bestimmten Latitude- und Longitude-Intervall liegt.

Der Grund hierfür ist leicht einzusehen: Wenn wir dem FS5 schon früh mitteilen, in welchen Grenzen unsere Szenerie liegt, muß er unsere Szenieriedatei auch nur dann lesen, wenn sie auch tatsächlich benötigt wird.

Legen wir dagegen die Extremgrenzen [W180..E180] und [S90..N90] fest, so wird unsere Szeneredatei immer gelesen, egal, ob sie nun wirklich verwendet wird.

In den meisten Fällen werden wir schon zu Beginn des Baus einer Szenerie wissen, welchen Bereich wir damit abdecken wollen, insbesondere dann, wenn wir die verschiedenen Landschaftstypen mit "TERRAIN" und "STRIPE" definieren. Sobald wir also die Grenzen unserer Szenerie kennen, sollten wir dem FS5 dies mitteilen. Dafür bietet das FS5DesignerTool den Befehl "RANGE". In ihm legen wir die Minimal- und Maximalwerte der Latitude und Longitude fest. Für unser Beispiel im letzten Abschnitt würde zum Beispiel genügen:

RANGE (N45,0,0), (N55,0,0), (E4,0,0), (E20,0,0)

Wenn wir keinen "RANGE"-Befehl benutzen, schreibt das FS5DesignerTool automatisch die Extremeinstellungen, also

RANGE (S90,0,0), (N90,0,0), (W180,0,0), (E180,0,0)

Beachten Sie bitte, daß die Einstellungen über "RANGE" immer nur einmal pro Szeneriebeschreibung gemacht werden können. *richtig ist: (E177,*

5.5 Gebirgetypen

Bisher haben wir nur den Basislandschaftstyp 0 (flaches Land) kennengelernt. Der FS5 bietet daneben noch den Typ 1 (für Berge) und die Typen 2 und 3 (See), die ebenfalls mit den Befehlen "TERRAIN" und "STRIPE" benutzt werden können.

Die Benutzung der See-Typen ist noch nicht abschließend untersucht worden. Wir werden daher nicht näher auf sie eingehen. Einige grundsätzliche Informationen zur Benutzung der See-Typen befinden sich in der Befehlsübersicht im Anhang.

Zur Demonstration der Definition von Bergen wollen wir einmal unser bekanntes "STRIPE"-Beispiel in eine Sequenz von Landschaftsblöcken aus Gebirge umwandeln.

```
STRIPE 42, 5, 9, 1,
      1000,1,1,0,0,
      1500,1,2,0,0,
      2000,1,3,0,0,
      3000,1,4,0,0,
      2000,1,5,0,0
```

An diesem Beispiel läßt sich gut erkennen: Nicht alle Berge des Landschaftstyps eines Blocks sind gleich hoch. Und, die Festlegung der Höhenwerte stellt nur eine Art "Grundhöhe" dar; die Berge erheben sich dann oberhalb dieser Grundhöhe. Die Höhenwerte sagen also nichts aus über die exakte Höhe der Berge.

Eine Auflistung der bisher erkannten Gebirgetypen befindet sich ebenfalls in der Befehlsübersicht im Anhang beim "TERRAIN" und "STRIPE"-Befehl.

Damit wollen wir die Definition der verschiedenen Landschaftstypen abschließen. In der praktischen Arbeit werden sich der "TERRAIN"- und "STRIPE"-Befehl sicher als hilfreich erweisen, um schnell große Gebiete zu definieren. Sollen dagegen z.B. Berge exakter und naturgetreuer definiert werden, empfiehlt es sich stattdessen, Polygone zu benutzen. Näheres dazu im Kapitel über die einfachen Grafikoperationen.

6. Flughäfen und Landebahnen

Sicher haben Sie bei der Benutzung des FS5 bei bestehenden Szenarien schon die Möglichkeit schätzen gelernt, über das "world"-Menü im Untermenü "Airports" unmittelbar angeben zu können, von welchem Flughafen aus Sie abfliegen wollen. Der FS5 übernimmt dann das Setzen der exakten Koordinaten des Flughafens, so daß Sie sich anschließend auf der Runway des entsprechenden Flughafens befinden.

6.1 AIRPORT und GEO_AREA

Mit dem FS5DesignerTool können Sie solche Einträge im "Airport"-Menü selbst erzeugen. Der entsprechende Befehl heißt "AIRPORT". Darin legen Sie die exakten Absolutkoordinaten des Flughafens fest; die Höhe des Flughafens dürfen Sie hier sogar mit Nachkommastellen angeben – sie wird mit einer Genauigkeit von 1/256 m gespeichert. Neben der Orientierung der Landebahn legen Sie den Flughafenamen fest, sowie die Frequenzen des Sprechfunks (COM1) und der beiden Navigationsempfänger (NAV1 und NAV2). Für beide Navigationsempfänger sind dann noch die Azimutkurse in grad (OBI1 und OBI2) anzugeben. Insgesamt lautet die Befehlssyntax:

```
AIRPORT <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>,  
      <Orientierung der Runway in Grad>, "<Flughafenname>",&br/>      <COM1-Frequenz in MHz>,  
      <NAV1-Frequenz in MHz>, <OBI1-Kurs in grad>,  
      <NAV2-Frequenz in MHz>, <OBI2-Kurs in grad>
```

Vielleicht ist Ihnen schon aufgefallen, daß im Airport-Menü des FS5 nicht nur die Flughäfen selbst stehen, sondern zusätzlich auch die Szeneriegebiete ("Geographical Areas"), in denen sich diese Flughäfen befinden. Wenn Sie das Szeneriegebiet wechseln, werden automatisch die dazu passenden Flughäfen angezeigt. Wollen Sie nun Ihren eigenen Flughafen definieren, so ist es erforderlich, dafür erst einmal ein eigenes Szeneriegebiet zu vereinbaren.

Das FS5DesignerTool erlaubt es, innerhalb eines Quelltextes maximal ein Szeneriegebiet zu erzeugen. Dies läßt sich mit dem Befehl "GEO_AREA" bewerkstelligen. Darin vergeben Sie einfach einen Namen für das Szeneriegebiet, also zum Beispiel über:

```
GEO_AREA "Ostwestfalen-Lippe"
```

Der verwendete Name darf nicht länger als 25 Zeichen sein.

Nachdem wir uns ein Szeneriegebiet definiert haben, können wir nun einen Flughafen angeben. Wir nehmen uns dazu wieder den Flughafen Paderborn-Lippstadt vor und definieren einen entsprechenden Eintrag im "Airport"-Menü. Beachten Sie bitte, daß die angegebenen Daten nur Beispieldaten sind.

```
AIRPORT (51,37,10.2), (8,37,43.8), 228.0,  
      239, "Paderborn Tower",  
      117.00,  
      168.0,0,  
      173.0,0
```

6.2 RUNWAY

Wenn wir uns schon an einen bestimmten Flughafen positionieren können, dann müssen wir diesen auch zumindest in seinen wesentlichen Elementen darstellen können. Dafür werden wir jetzt den "RUNWAY"-Befehl kennenlernen; dieser stellt zwar eine primitive Grafikoperation dar und hätte eigentlich schon im Kapitel über primitive Operationen abgehandelt werden können. Daß wir dies dennoch erst jetzt tun, liegt an der Komplexität des Befehls. Sehr viele Details einer Landebahn können darin spezifiziert werden:

```
RUNWAY <Latitude in (G,M,S)>, <Longitude in (G,M,S)>,  
      <Altitude in m>, <Orientierung in grad>,  
      <Länge in Fuß>, <Breite in Fuß>,  
      <ID aus [0..255]>, <Oberfläche aus [0..3]>,  
      <Markers aus [0..255]>, <Lights aus {0,1,4,5}>,  
      <End 1 Schwellenmarkierungen aus {0,1}>,  
      <End 1 Länge des Markierungsbereichs Typ 1 in Fuß>,  
      <End 1 Länge des Markierungsbereichs Typ 2 in Fuß>,  
      <End 1 Anflugbefeuerung aus {2,6,7}>,  
      <End 1 Strobezahl>,  
      <End 2 Schwellenmarkierungen aus {0,1}>,  
      <End 2 Länge des Markierungsbereichs Typ 1 in Fuß>,
```



```

<End 2 Länge des Markierungsbereichs Typ 2 in Fuß>,
<End 2 Anflugbefeuerung aus {2,6,7}>,
<End 2 Strobezahl>,
<VASIS 1 aus {0,1,2}>, <Winkel VASIS 1 in grad>,
<VASIS 2 aus {0,1,2}>, <Winkel VASIS 2 in grad>

```

Wir wollen diese der Reihe nach durchgehen. Die ersten sechs Parameter bedürfen wohl keiner Erklärung, hier werden Position, Orientierung sowie die Ausmaße der Landebahn festgelegt.

Im Parameter "ID" legen wir fest, ob die Runway mit einer Landebahnnummer im Bereich [1..63] beschriftet werden soll. Optional kann dazu auch einer der Buchstaben "L", "C" oder "R" auf die Landebahn gezeichnet werden. Für "L" addieren Sie dann den Wert 64 zur oben gewählten Zahl; für "R" addieren Sie die 128, und für "C" die 192. Das Ergebnis schreiben Sie in den Parameter "ID". Soll keine Landebahnnummer gezeichnet werden, wählen Sie die "0".

Anschließend definieren wir das Aussehen der Landebahnoberfläche. 4 Typen stehen im Parameter "Oberfläche" zur Verfügung.

Wert	Typ
0	graues Muster
1	betonartige Oberfläche, mit Punkten
2	betonartige Oberfläche, dunkler als 1, mit Streifen
3	grasartige Oberfläche

Zusätzlich aufgemalte Markierungen auf der Landebahn beschreiben wir im Parameter "Markers". Wenn Sie mehrere Elemente benutzen wollen, so addieren Sie einfach die einzelnen Wertigkeiten und schreiben das Ergebnis in den Parameter "Markers". Lediglich die Typen "8" und "64" dürfen nicht kombiniert werden, da sie sich gegenseitig ausschließen.

Wert	Bedeutung
1	Seitenlinien
2	Schwellenmarkierungen
4	Landemarkierungen (3 schmale Linien)
8	Abstandsmarkierungen (breite Linien)
16	Mittellinie
32	Darstellung der ID-Zahl
64	Abstandsmarkierungen (schmale Linien)

Schließlich besteht im Parameter "Lights" die Möglichkeit festzulegen, wie die Landebahn beleuchtet sein soll:

Wert	Bedeutung
0	keine Beleuchtung
1	Beleuchtung an den Seiten
4	Beleuchtung in der Mitte
5	Beleuchtung an den Seiten und in der Mitte

Die nun nachfolgenden Parameter müssen für beide Enden der Landebahn ("End1" und "End2") getrennt vereinbart werden. Dazu gehören Schwellenmarkierungen, Markierungsbereiche, Anflugbefeuerungen und Stroboskoplichter.

Wenn Sie Schwellenmarkierungen wollen, so stellen Sie den entsprechenden Parameter auf "1", sonst auf "0".

Weiter können Sie zwischen zwei Sorten von Markierungsbereichen an jedem Ende der Landebahn auswählen. Soll ein Ende der Landebahn mit einem der beiden möglichen Markierungsbereiche versehen werden, so können Sie diesen über eine Länge verschieden von Null anwählen. Beide Sorten von Markierungsbereichen dürfen Sie allerdings nicht an einem Ende verwenden, sie schließen sich gegenseitig aus. Bei Verwendung dieser Längenparameter für die Markierungsbereiche konnte während der umfangreichen Tests des FS5DesignerTools festgestellt werden, daß der FS5 nur dann stabil arbeitete, wenn als Länge der Wert "1010" (bzw. "0" für keinen Markierungsbereich) eingestellt wurde. Andere Werte führten zu undefiniertem Verhalten des FS5.

Wünschen Sie eine Anflugbefeuerung, so können Sie im entsprechenden Parameter unter drei verschiedenen Typen wählen:

Wert	Typ
2	Anflugbefeuerung mittlerer Intensität (MALSR)
6	ALSF-1
7	ALSF-2

Die Anzahl der Blitzlichter der Anflugbefeuerung legt der Parameter "Strobezahl" fest.

Wir sind nun fast am Ende der Parameterliste angelangt. Zuletzt ist noch zu entscheiden, ob die Landebahn mit einem Sichtflug-Gleitwinkelanzeigesystem (VASIS) versehen sein soll. Dies legen wir fest über die Anzahl der Lichter neben der Landebahn: *Wert Anzahl VASIS-Balken*

0	0
1	2
2	3

Die Ausrichtungen des VASIS werden in den Winkelparametern des VASIS beschrieben.

So, nun wollen wir uns endlich ein Beispiel zum "RUNWAY"-Befehl ansehen. Wir definieren eine Landebahn für den Flughafen Paderborn-Lippstadt, damit wir den "AIRPORT"-Befehl vom Beginn dieses Kapitels gleich sinnvoll zum Positionieren auf unserer Landebahn verwenden können.

```

RUNWAY (51,36,55) , (8,37,2) , 228,      ' Koordinaten
239,                                     ' Orientierung
6812,281,                                ' Ausmaße
216,                                     ' ID = Nr.24 + 192("C")
1,                                       ' Oberflächentyp
119,                                    ' Markers
5,                                       ' Lights
1,                                       ' End1
1010,
0,
7,
5,
1,                                       ' End2
1010,
0,
2,
3,
0,0,                                    ' kein VASIS
0,0

```

Vergleichen Sie doch einmal bei dieser Landebahn die Darstellung während der hellen Tagesstunden, und die Darstellung nachts.

Der "RUNWAY"-Befehl nimmt unter den einfachen Operationen eine gewisse Sonderrolle ein; Sie brauchen bei Definition einer Runway keinen Referenzpunkt zu definieren, wenn im betreffenden Sichtbarkeitsbereich keine anderen einfachen Operationen verwendet werden.

6.3 ATIS

Abschließend wollen wir uns in diesem Kapitel der automatischen Flugplatzinformation, kurz ATIS zuwenden. Das FS5DesignerTool erlaubt die Definition dieser Informationseinrichtung für jeden Flugplatz. Im Befehl "ATIS" können Sie neben Festlegung der notwendigen Informationen über Koordinaten, Frequenz und Reichweite die Nummern von maximal vier bevorzugten Landebahnen angeben, über die der Flugplatz verfügt. Und, natürlich, können Sie die ATIS-Meldung selbst definieren, das ist ein Text, der in einem kleinen Fenster des FS5-Bildschirms auftaucht.

Die vollständige Befehlssyntax lautet:

```

ATIS <Frequenz in MHz>, <Reichweite in 2048m-Einheiten>,
    <Latitude in (G,M,S)>, <Longitude in (G,M,S)>,
    <Runway1>, <Runway2>, <Runway3>, <Runway4>,
    "<ATIS Message>"

```

In der ATIS Message legen Sie den expliziten Text fest, der auf dem FS5-Bildschirm angezeigt werden soll. Bitte beachten Sie, daß in diesem String nur Zeichen im Bereich ASCII 32 bis ASCII 127 unmittelbar verwendet werden können. Insbesondere die deutschen Umlaute 'Ä'/ä, 'Ö'/ö, 'Ü'/ü' sowie das scharfe 'ß' liegen oberhalb dieser Grenze und können also nicht verwendet werden; der FS5 erzeugt dann undefinierte Zeichenfolgen.

Eine weitere Besonderheit der ATIS Message besteht darin, daß für die ASCII-Codes 128 bis 148 vordefinierte Strings eingesetzt werden. Steht etwa in Ihrer ATIS Message das Zeichen "ç" (ASCII 135), so wird nicht dieses Zeichen, sondern der Text " ADVISE CONTROLLER " ausgegeben. Sie können auf Ihrer PC-Tastatur jedes Zeichen im Bereich ASCII 0 bis ASCII 255 direkt angeben, indem Sie die "Alt"-Taste festhalten und dabei den Code des gewünschten Zeichens auf dem Ziffernblock Ihrer Tastatur eingeben.

Hier eine Übersicht über die vordefinierten Texte der ATIS Message:

Code	Text
128	" Weather - "
129	" Observation "
130	" <aktuelle Stunde>:00 ZULU "
131	""
132	" Temperature 25 - "
133	" Information "
134	" Landing and departing runway <Index der 1. Runway> "
135	" Advise controller "
136	" Altimeter 29.95 - "
137	" Visibility 10 - "
138	" Winds <Windrichtung aus dem "Wind Menu"> "
	(Falls im "Wind Menu" keine Windrichtung definiert wurde, wird dieses Zeichen ignoriert.)
139	" Measured ceiling "
140	" on initial contact "
141	""
142	""
143	"Microsoft Flight Simulator"
144	" requesting "
145	"clearance "
146	" , you are cleared "
147	" ... "
148	"7777 "

Wir geben einmal ein (fiktives) Beispiel an:

ATIS 123.2,	'Frequenz in MHz
40,	'Reichweite 80km
(N51,36,55), (E8,37,2),	'Latitude, Longitude
24,0,0,0,	'Runway-Nummern 1 bis 4
"Paderborn/Lippstadt à - éâ"	'Codes 133,130,134

Natürlich lassen sich auch umfangreichere ATIS-Meldungen realisieren. Das Beispiel soll nur demonstrieren, mit wie einfachen Mitteln Sie auf die vordefinierten Meldungen des FS5 zugreifen können.

7. Ablaufstrukturen

Bisher haben wir einzig Unterprogramme und Sprünge als strukturierende Elemente für FS5DesignerTool-"Programme" kennengelernt. Diese sind in der Praxis aber noch nicht allzu nützlich, weil beispielsweise die Beschränkung auf Sichtbarkeitsbereiche existiert. Und, wir können damit nur Einfluß nehmen auf die einfachen Grafikoperationen, haben also keine Hilfen, beispielsweise für die verschiedenen Navigationselemente (VOR, ILS etc.).

Auch kommt es häufig vor, daß man ein bestimmtes Objekt an mehreren Stellen verwenden möchte. Oder, ein Objekt wird mit nur leicht variiertem Aussehen an verschiedenen Orten plziert. Beispiel dafür ist etwa ein Gebäude, das mit dem POLYGON-Befehl "gebaut" wurde. Hier genügt es ja häufig, die Grundparameter Länge, Breite und Höhe anzugeben, und vielleicht eine Farbe. Gerade hier wäre also eine Unterstützung von "Parametern" wünschenswert.

7.1 Der "PROC"-Mechanismus -Der intelligente Kopierbefehl-

An dieser Stelle bietet das FS5DesignerTool eine leicht zu verwendende Hilfe an: Sie können Teile Ihres Programmtexes so vereinbaren, daß der vereinbarte Code an mehreren Stellen explizit "aufgerufen" werden kann. Und, Sie können den vereinbarten Code mit Parametern "aufrufen", das heißt, ein solches Codestück kann an verschiedenen Stellen in der Szenerie unterschiedliche Wirkungen haben. Anknüpfend an unser Gebäude-Beispiel können Sie etwa die Grundparameter des Gebäudes angeben und gleich bei der Vereinbarung der Eckpunkte des Gebäudes auf diese Grundparameter zugreifen.

Wir geben einmal ein einfaches Beispiel.

```
PROC "Haus", 3
  DEFINE_POINTS 8, 0,
    0, 0, 0, 'Punkt links unten vorne
    %1, 0, 0 'Punkt links unten hinten
    %1, %2, 0, 'Punkt rechts unten hinten
    0, %2, 0, 'Punkt rechts unten vorne
    0, 0, %3, 'Punkt links oben vorne
    %1, 0, %3, 'Punkt links oben hinten
    %1, %2, %3, 'Punkt rechts oben hinten
    0, %2, %3 'Punkt rechts oben vorne
  SHADE_COLOUR 10 'Farbe hellbraun
  POLYGON 4, 0, 0, 0, 0, 4, 7, 3 'Fläche vorne
  POLYGON 4, 0, 0, 0, 3, 7, 6, 2 'Fläche rechts
  POLYGON 4, 0, 0, 0, 2, 6, 5, 1 'Fläche hinten
  POLYGON 4, 0, 0, 0, 0, 1, 5, 4 'Fläche links
  POLYGON 4, 0, 0, 0, 4, 5, 6, 7 'Fläche Dach
ENDPROC

CALLPROC "Haus", 30, 100, 200
```

Wir haben zuerst in der Sequenz von "PROC" bis "ENDPROC" eine Codefolge zum Zeichnen eines Hauses definiert. Wohlgermerkt, diese Sequenz alleine bewirkt noch nicht, daß irgendetwas auf dem FS5-Bildschirm passiert! Sie muß erst mit "CALLPROC" aufgerufen werden. Aber der Reihe nach.

Zu Beginn der "PROC" teilen wir dem FS5DesignerTool mit, unter welchem Namen er diese Codesequenz ablegen soll. Die Sequenz wird als "Haus" vereinbart und erhält drei Parameter (zweiter Parameter in "PROC"). Der Mechanismus, mit dem die Parameter in der Codesequenz benutzt werden, ist den meisten Lesern sicher schon bekannt. Er ähnelt der Verwendung der Parametern in Batch-Dateien unter MS-DOS. Die Parameter werden von 1 beginnend hochgezählt; der erste Parameter wird also symbolisch durch %1 ersetzt, der zweite durch %2 und so weiter.

In unserem Beispiel sind die Parameter 1, 2 und 3 die Breite, Länge und Höhe des Gebäudes. Wir hätten aber beispielsweise auch noch die Farbe als Parameter vereinbaren können. (Dann müßte es "PROC "Haus", 4" lauten, und "SHADE_COLOUR %4:"). Oder, wir könnten einen Drehwinkel für das Haus definieren, indem wir anstatt "DEFINE_POINTS" den ähnlichen Befehl "DEFINE_ROTATED_POINTS" verwenden, und den Drehwinkel als Parameter 5 übergeben.

Nicht als Parameter vereinbaren können wir Strings (Zeichenketten). Der "PROC"-Mechanismus erlaubt zur Zeit nur numerische Ausdrücke als Parameter.

Im Prinzip können Sie innerhalb einer "PROC" sämtliche FS5DesignerTool-Befehle verwenden; nicht erlaubt sind nur "END", "PROC" und "CALLPROC". Daß dies so ist, hängt mit einer Einschränkung zusammen: **In der aktuellen Version des FS5DesignerTools ist es nicht möglich und erlaubt, "PROC"s zu schachteln.** Sie dürfen also **nicht** schreiben:

```
PROC "1"
...
    PROC "2"
    ...
    ENDPROC
...
ENDPROC
```

Ebenso darf innerhalb einer "PROC" keine andere "PROC" verwendet werden; nicht erlaubt ist also auch:

```
PROC "1"
...
    CALLPROC "2"
    ...
ENDPROC
```

"PROC"s stellen nämlich keine echten Unterprogramme dar, sondern sie werden an ihren "Aufrufstellen" textuell in den Code eingefügt. Wenn Sie also zweimal 'CALLPROC "Haus"...' in Ihre Szeneriebeschreibung einfügen, so ist der resultierende Code genauso lang, als hätten Sie die zwei Häuser explizit angegeben.

Trotz dieser beiden Einschränkungen halten wir die "PROC"s für ein geeignetes Mittel, den Aufwand bei der Erzeugung großer Szenarien so gering wie möglich zu halten. Außerdem ist der resultierende Programmtext weitaus übersichtlicher und besser "lesbar" als explizit ausprogrammierter Code.

In zukünftigen Versionen des FS5DesignerTools wird es wahrscheinlich auch möglich sein, "PROC"s geschachtelt zu verwenden, so daß dann eine Einschränkung in der Flexibilität der "PROC"s wegfällt. Dann ist sogar Rekursion möglich.

Für den Augenblick läßt sich der "PROC"-Mechanismus schon einmal dazu verwenden, um bestimmte Objekte, die leicht parametrisierbar sind, im Vorhinein zu definieren, um auf diese dann einfach zuzugreifen. Zusammen mit der Möglichkeit der Verwendung von einfachen numerischen Ausdrücken lassen sich so auch komplexere "Grundobjekte" bauen.

7.2 Der "INCLUDE"-Mechanismus *(Einbinden aus anderen Dateien)*

Haben Sie einmal mit "PROC" einige Grundobjekte zusammen, so müssen Sie diese nicht alle in Ihrer Quelldatei für das FS5DesignerTool vereinbaren, sondern Sie können diese quasi als "Bibliothek" in eine andere Datei auslagern, um diese nur bei Bedarf einzubinden. Auf diese Weise können Sie aus verschiedenen Szenarien immer wieder auf dieselben Objekte zugreifen.

Der Befehl, den Sie hierfür verwenden können, heißt "INCLUDE". Haben Sie also etwa in der Datei "objekte.fst" verschiedene Objekttypen durch "PROC" vereinbart, so schreiben Sie einfach am Beginn Ihrer Szeneriebeschreibung den Befehl

```
INCLUDE "objekte.fst"
```

Im Prinzip genügt es, den "INCLUDE"-Befehl erst unmittelbar vor dem Aufruf einer in "objekte.fst" definierten "PROC" einzufügen. Aus Gründen der Übersichtlichkeit empfiehlt sich aber die Verwendung gleich zu Beginn der Szeneriebeschreibung.

Beachten Sie auch bitte immer, daß alle Dateien, die Sie mit "INCLUDE" einbinden wollen, mit einem "END" abgeschlossen sein müssen. Das FS5DesignerTool bricht seine Arbeit sonst vorzeitig ab.

Wenn Sie mögen, können Sie auch ganze ablauffähige Programmteile (also nicht nur "PROC"s) in "INCLUDE"-Dateien auslagern. Wenn Sie diese Dateien dann inkludieren, so wird der Code an genau der Stelle eingefügt, an der der "INCLUDE"-Befehl steht. Auf diese Weise können Sie die Übersichtlichkeit Ihrer Szeneriebeschreibung weiter steigern.

Vermeiden Sie bitte eine Schachtelung von "INCLUDE"s. Hier gilt dasselbe wie beim "PROC"-Befehl.

8. Die FS5 Szeneriebibliothek - Bedeutung der IDs

Wenn Sie beim FS5 in der Menüzeile den Eintrag "Scenery" anklicken, und dann das Untermenü "Scenery Library" anwählen, werden Ihnen verschiedene "Welten" präsentiert. Die bekanntesten darunter sind die "FS5 Welt" und die "Sprühflugzeugwelt". Vielleicht ist Ihnen auch aufgefallen, daß neben jeder dieser "Welten" eine Datei mit der Extension "vis" steht, z.B. "world.vis" oder "cropw.vis".

Daß der FS5 verschiedene "Welten" kennt, hat unter anderem den Grund, daß sich verschiedene Szenieriedefinitionen gegenseitig überschneiden und sogar überdecken können, wenn sie im selben Gebiet definiert sind. Dies kann zum Beispiel dann passieren, wenn an irgendeiner Stelle auf der Welt eine neue, detailliertere Szene erstellt werden soll; diese wird sich in den meisten Fällen mit der "FS5 Welt" überschneiden, da diese ja den gesamten Erdball abdecken soll.

Genau für diesen Fall wurde der FS5 mit einem Mechanismus versehen, um dennoch mehrere Szenerien für eine bestimmte Stelle definieren zu können. Jede Szeneriebeschreibung wurde daher mit einer Ident-Nummer versehen. Den ID-Befehl kennen Sie ja schon aus dem Kapitel "Erste Schritte". Die "FS5-Welt" besteht beispielsweise nur aus Szeneriebeschreibungen, deren Ident-Nummern in der Menge {1,3,4} liegen.

Das heißt: Haben Sie eine Szene erzeugt und diese mit einer dieser Ident-Nummern versehen, so wird diese automatisch geladen, wenn die "FS5 Welt" eingestellt wurde. Wir werden nun zeigen, wie wir einen eigenen Eintrag in der FS5-Szeneriebibliothek erzeugen können.

Der Befehl, den wir dafür verwenden können, heißt "SCENERY". Darin legen Sie maximal fünf verschiedene Szenerietypen fest, die der entsprechenden "Welt" angehören sollen. Sollen weniger Typen vereinbart werden, so bleiben die undefinierten auf "0":

```
SCENERY 18,0,0,0,0,"Aerosoft Welt",
```

"Dies ist die Aerosoft Welt. Testen Sie die Möglichkeiten des FS5DesignerTools!"

In diesem Beispiel haben wir die "Aerosoft Welt" mit Szenieriedateien des Typs "18" verknüpft. Die übrigen Zahlenwerte sind undefinierte Szenerietypen (d.h. die "Aerosoft Welt" besteht ausschließlich aus Szenerien des Typs "18"). In den letzten beiden Parametern vereinbaren Sie den Namen der "Welt" und geben eine Kurzbeschreibung der "Welt".

Setzen Sie den "SCENERY"-Befehl einfach an den Anfang Ihrer Szeneriebeschreibung. Wenn Sie das FS5DesignerTool dann von DOS aus starten, erhalten Sie als Ausgabe neben der eigentlichen Szeneriebeschreibung (standardmäßig "fsout.bg1") eine zusätzliche Datei mit der Extension "vis" (standard: "fsout.vis"). Kopieren Sie diese wie die Szenieriedatei (Extension ".bg1") in das "scenery"-Directory des FS5. Sie erhalten dann einen Eintrag in der "Scenery Library".

Vergessen Sie aber nicht, auch Ihre Szeneriebeschreibung durch Setzen der "richtigen" Ident-Nummer der richtigen "Welt" zuzuordnen. Anderenfalls werden Sie Ihre Szene vielleicht in einer anderen "Welt" wiederfinden.

9. Speicherprobleme

Das FS5DesignerTool kann als DOS-Anwendung theoretisch **bis zu 1MB Speicher für Daten** adressieren. Bei kleinen Systemen stehen immer noch einige hundert Kilobytes zur Verfügung, die für die meisten Anwendungen ausreichen sollten.

Bei sehr großen Quelltexten kann es jedoch passieren, daß das FS5DesignerTool die Eingabe nicht mehr verarbeiten kann, weil der verfügbare Speicher nicht mehr ausreicht. Ebenso kann bei der konsequenten Verwendung von PROCs der Speicherplatz ausgehen; dies liegt insbesondere daran, daß ein PROC-Programmstück nicht wirklich als Unterprogramm übersetzt wird, sondern stattdessen textuell in den Quelltext eingefügt wird. Ein zweimaliges Verwenden einer PROC entspricht also derselben Quelltextlänge wie das direkte Angeben des impliziten "PROC"-Programmtextes.

Haben wir also den erlaubten Speicherplatz überschritten, dann muß Abhilfe geschaffen werden. In diesem Fall kommt uns der FS5 mit einer Eigenschaft entgegen, die er selbst in einigen der bereits existierenden Szenarien ausnützt. Wie bereits im Kapitel "FS5 Szeneriebibliothek" erwähnt, lädt der FS5 bei Vorgabe eines Bibliothekseintrags (standardmäßig ist dies "world.vis") automatisch alle Szenerie-dateien ein, deren ID-Nummer mit denen im Bibliothekseintrag übereinstimmen. (Bei "world.vis" sind dies beispielsweise alle Szenerie-dateien mit ID-Werten "1", "3", "4", und "5".)

Wir können unser Speicherproblem also einfach dadurch umgehen, daß wir unseren zu großen Quelltext in zwei oder mehr Quelltexte aufspalten und diese getrennt übersetzen. Wenn alle Quelltexte dieselbe ID-Nummer erhalten (Befehl "ID"), werden sie automatisch zum richtigen Zeitpunkt geladen.

Wir können unser Problem aber auch anders lösen. Überprüfen Sie doch einmal, wieviele "Hintergrundprogramme" Sie im Augenblick der Übersetzung im Speicher stehen haben. (Unter MS-DOS steht dafür der Befehl "mem" zur Verfügung.) Entfernen Sie alle überflüssigen Anwendungen aus dem Hauptspeicher (meistens gleichbedeutend mit Zurücksetzen des Systems) und versuchen Sie erneut, Ihren Quelltext zu übersetzen.

Eine dritte Möglichkeit, die sich bei der Verwendung von sehr vielen einfachen Operationen anbietet, ist die Verwendung von echten Unterprogrammen. Das FS5DesignerTool bietet ja die Verwendung von Unterprogrammsprüngen mit CALL, d.h. wenn Sie Ihren Code daraufhin "optimieren", daß sich häufig wiederholende Sequenzen nicht einfach durch Verwendung von PROCs wiederholen, sondern nur einmal als Unterprogramm angegeben werden, können Sie Speicherplatz sparen.

Vergessen Sie dabei bitte nicht, daß Unterprogrammsprünge beim FS5 immer Vorwärtssprünge sein müssen, d.h. Sie können nur Code aufrufen, der auf die Aufrufstelle folgt.

Anhang

A. Alphabetische Liste der FS5DesignerTool-Befehle

- AIRPORT -

Definiert einen Eintrag in der Airport-Liste des FS5. In diesem Befehl kann der Altitude-Wert als Fließkommazahl eingegeben werden; sie wird in Einheiten von 1/256 m gespeichert. In jedem Fall muß vor dem ersten AIRPORT-Befehl einmal der Befehl "GEO_AREA" benutzt werden; die definierten Flughäfen werden dann der entsprechenden "Gegend" (engl. "Area") zugeordnet. Befehlsformat:

AIRPORT <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>,
 <Orientierung der Runway in Grad>, "<Flughafenname>",
 <COM1-Frequenz in MHz>,
 <NAV1-Frequenz in MHz>, <OBI1-Kurs in grad>,
 <NAV2-Frequenz in MHz>, <OBI2-Kurs in grad>

- AREA_TEST -

Legt für einen zusammenhängenden Teil der primitiven Operationen als einen zusammenhängenden Sichtbarkeitsbereich fest. Neben der absoluten Koordinate ist die Größe des Bereichs, nach oben abgeschätzt, anzugeben; dieser "Range"-Wert ist in Einheiten von 2048 m anzugeben. Zum korrekten Abarbeiten der Szenieriefles ist bei Verwendung von primitiven Operationen unbedingt dieser Befehl zu benutzen, da der FS5 die Szenerie sonst gar nicht lesen kann. Befehlsformat:

AREA_TEST <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Range aus [1..255]>

- ATIS -

Legt für einen Flughafen eine automatische Flughafeninformation (ATIS) fest. Erlaubte Werte für die Frequenz sind 100.00 bis 199.50; hier gilt ein Raster von 0.05 MHz = 50 KHz. In den Parametern "Runway1" bis "Runway4" sind die Nummern von bevorzugten Landebahnen anzugeben; sind weniger als vier Landebahnen vorhanden, sind die übrigen "Runway"-Parameter auf "0" zu setzen. Im Parameter "ATIS Message" dürfen nur Zeichen im Bereich ASCII 32 bis 127 verwendet werden, also insbesondere keine deutschen Umlaute.

Zusätzlich werden Zeichen im Bereich ASCII 128 bis ASCII 148 entsprechend folgender Tabelle in Strings übersetzt:

Code	Text
128	" Weather - "
129	" Observation "
130	" <aktuelle Stunde>:00 ZULU "
131	""
132	" Temperature 25 - "
133	" Information "
134	" Landing and departing runway <Index der 1. Runway> "
135	" Advise controller "
136	" Altimeter 29.95 - "
137	" Visibility 10 - "
138	" Winds <Windrichtung aus dem "Wind Menu"> " (Falls im "Wind Menu" keine Windrichtung definiert wurde, wird dieses Zeichen ignoriert.)
139	" Measured ceiling "
140	" on initial contact "
141	""
142	""
143	"Microsoft Flight Simulator"
144	" requesting "
145	"clearance "
146	" , you are cleared "
147	" ... "
148	"7777 "

Die ATIS Message darf nicht länger 255 Zeichen sein. Befehlsformat:

ATIS <Frequenz in MHz>, <Reichweite in 2048m-Einheiten>,
 <Latitude in (G,M,S)>, <Longitude in (G,M,S)>,
 <Runway1>, <Runway2>, <Runway3>, <Runway4>, "<ATIS Message>"

- AUTO_CALL -

Legt die unmittelbar nachfolgende Sequenz von primitiven Operationen als echtes Unterprogramm ab, das mit RETURN" abzuschließen ist. Dadurch kann der FS5 komplizierte Sequenzen schneller abarbeiten.

AUTO_CALL ist eine einfache Operation. Befehlsformat:

AUTO_CALL (ohne Parameter)

- BEGIN_SURFACE -

Beginnt die Definition einer großen Bodenfläche. BEGIN_SURFACE ist eine einfache Operation. Befehlsformat: BEGIN_SURFACE (ohne Parameter)

- BITMAP -

Spezifiziert für nachfolgende primitiven Operationen ein Bitmap-File. Dieses File muß im Texture-Directory des FS5 stehen. Kann zum Füllen von Polygonen mit Bitmaps verwendet werden. Im Skalierungsfaktor kann die Auflösung der Bitmap spezifiziert werden; 0 bedeutet starke Vergrößerung, bei größeren Werten wird die Bitmap verkleinert. Die Delta-East und Delta-North-Koordinaten ermöglichen eine Verschiebung der Bitmap-Darstellung. Die Dummy-Parameter "Dummy1" und "Dummy2" sind für ein korrektes Funktionieren auf "0" zu setzen. BITMAP ist eine einfache Operation. Befehlsformat:

BITMAP "<Name der Füllmuster-Datei, mit Extension!>",

<Delta-North>, <Delta-East>, <Skalierungsfaktor aus [0..32767]>, <Dummy1>, <Dummy2>

- BLINK -

Setzt einen gelb blinkenden Punkt am vorher spezifizierten Referenzpunkt. Kann als Orientierungshilfe dienen. BLINK ist eine einfache Operation. Befehlsformat:

BLINK (ohne Parameter) *→ Text vorgegebener POINT*

- BUILDING -

Erzeugt Gebäude an Relativposition zum vorher spezifizierten Referenzpunkt. In den Parametern "Lat.-Differenz" und "Long.-Differenz" ist die Länge und Breite des Gebäudes in m anzugeben. Im Parameter "Stockwerke" ist die Anzahl der Stockwerke anzugeben. Im Parameter "Fenster" können 7 verschiedene Muster angegeben werden:

Wert	Bedeutung
0	Beige mit vertikalen Streifen
1	Weiß mit vertikalen Streifen
3	Dunkelgrau mit vertikalen Streifen
4	Weiß mit dunklen vertikalen Streifen
5	Weiß mit horizontalen Streifen
6	Beige mit horizontalen Streifen
7	Hellgrau mit horizontalen Streifen

Weiter können im Parameter "Fenster" durch Addition von Wertigkeiten weitere Effekte erzielt werden:

Wert	Bedeutung
8	abgeschrägtes Dach
24	8 Wände anstatt 4

Im Parameter "Wände" kann festgelegt werden, aus wievielen Wänden das Gebäude bestehen sollen; ein gesetztes Bit in der Binärdarstellung dieses Parameters entspricht einer darzustellenden Wand. Bei Setzen der unteren 5 Bits (Wert 31) wird ein viereckiges Gebäude mit Dach dargestellt. Bei Setzen weiterer Bits wird immer eine Wand hinzugenommen (Wertigkeiten der einzelnen Bits: 32, 64, 128, 256). Die Wertigkeiten der Bits sind aufzuaddieren.

Wird also in "Fenster" durch Setzen der Bits mit Wertigkeiten "8" und "16" (=24 -> 8 Wände) auf 8 Seitenwände geschaltet, so ist "Wände" auf binär "1 1111 1111" = 511 zu setzen. Ein sicherer Wert ist "-1" =

"1111111111111111"; hier sind alle Wände erlaubt. BUILDING ist eine einfache Operation. Befehlsformat:

BUILDING <rel.Latitude in m>, <rel.Longitude in m>, <rel.Altitude in m>,
<Lat.-Differenz in m>, <Long.-Differenz in m>, <Stockwerke>,
<Fenster-"Muster" aus [0..7]>, <Wände>

- CALL -

Ermöglicht das Aufrufen eines Unterprogramms aus primitiven Operationen, das mit "RETURN" abzuschließen ist. CALL ist eine einfache Operation. Befehlsformat:

CALL "<Name der Marke>"

- CALLPROC -

Fügt spezifizierte Codesequenz textuell in den Quelltext ein. Es können nur Zahlen als Parameter verwendet werden, die in der entsprechenden PROC...ENDPROC-Sequenz vereinbart worden sein müssen. Es müssen genau so viele Parameter eingesetzt werden wie in der Vereinbarung. Die "aufgerufene" Codesequenz muß im Quelltext vorher vereinbart worden sein. Der Name der "aufgerufenen" Codesequenz muß identisch sein mit der Vereinbarung des Namens in PROC...ENDPROC, insbesondere hinsichtlich Groß-/Kleinschreibung. Befehlsformat:

CALLPROC "<Name>" <Parameterliste, getrennt durch Komma>

- COLOUR -

Setzt einen Farbwert für nachfolgende Zeichenoperationen. COLOUR ist eine einfache Operation. Befehlsformat:

COLOUR <Wert aus [0..255]>

- COLOUR_BRIGHTNESS -

Mit diesem Befehl ist es möglich, die Farbintensität zu setzen. COLOUR_BRIGHTNESS ist eine einfache Operation. Befehlsformat:

COLOUR_BRIGHTNESS <Wert aus [0..65535]>

- CONCAVE_POLY -

Zeigt an, daß die nachfolgenden Polygone konkav sind. Muß vor der ersten betreffenden "POLYGON"-Operation bzw. vor einer "BEGIN_SURFACE"... "END_SURFACE"-Sequenz stehen. CONCAVE_POLY ist eine einfache Operation. Befehlsformat:

CONCAVE_POLY (ohne Parameter)

- DEFINE_POINTS -

Vereinbart eine Liste von Punkten für nachfolgende einfache Operationen (z.B. MOVE_TO_POINT, POLYGON). Jeder Punkt wird mit einer Nummer abgelegt und besteht aus Latitude-, Longitude- und Altitude-Wert in m, wobei sich diese Werte immer auf den vorher gesetzten Referenzpunkt beziehen. Die Indices werden mit dem anzugebenden Startindex beginnend hochgezählt. DEFINE_POINTS ist eine einfache Operation. Befehlsformat:

```
DEFINE_POINTS <Anzahl der Punkte>, <Nr. des ersten Punktes>,  
    <rel.Latitude in m P1>, <rel.Longitude in m P1>, <rel.Altitude in m P1>,  
    <rel.Latitude in m P2>, <rel.Longitude in m P2>, <rel.Altitude in m P2>,  
    ...  
    <rel.Latitude in m Pn>, <rel.Longitude in m Pn>, <rel.Altitude in m Pn>
```

- DEFINE_ROTATED_POINTS -

Vereinbart eine Liste von Punkten für nachfolgende einfache Operationen (z.B. MOVE_TO_POINT, POLYGON). Die Punkte werden um einen anzugebenden Winkel um den vorher definierten Referenzpunkt gedreht definiert. Jeder Punkt wird mit einer Nummer abgelegt und besteht aus Latitude-, Longitude- und Altitude-Wert in m, wobei sich diese Werte immer auf den vorher gesetzten Referenzpunkt beziehen. Die Indices werden mit dem anzugebenden Startindex beginnend hochgezählt. DEFINE_ROTATED_POINTS ist eine einfache Operation. Befehlsformat:

```
DEFINE_ROTATED_POINTS <Anzahl der Punkte>, <Nr. des ersten Punktes>, <Drehwinkel aus [0..360]>,  
    <rel.Latitude in m P1>, <rel.Longitude in m P1>, <rel.Altitude in m P1>,  
    <rel.Latitude in m P2>, <rel.Longitude in m P2>, <rel.Altitude in m P2>,  
    ...  
    <rel.Latitude in m Pn>, <rel.Longitude in m Pn>, <rel.Altitude in m Pn>
```

- DOTTED_LINE -

Erzeugt eine gestrichelte Linie. Start- und Endpunkt sind in Relativkoordinaten zum vorher definierten Referenzpunkt anzugeben. DOTTED_LINE ist eine einfache Operation. Befehlsformat:

```
DOTTED_LINE <rel.Latitude in m P1>, <rel.Longitude in m P1>, <rel.Altitude in m P1>,  
    <rel.Latitude in m P2>, <rel.Longitude in m P2>, <rel.Altitude in m P2>,  
    <Anzahl der Teillinien für die Verbindung P1-P2>
```

- END -

Muß am Ende jedes Programmtextes, auch jedes INCLUDE-Files stehen, damit das FS5DesignerTool seine Arbeit ordnungsgemäß beendet. Befehlsformat:

END (ohne Parameter)

- ENDPROC -

Schließt eine durch PROC definierte Codesequenz ab. Befehlsformat:

ENDPROC (ohne Parameter)

- END_SURFACE -

Schließt eine Oberflächendefinition ab, die mit BEGIN_SURFACE begonnen wurde. END_SURFACE ist eine einfache Operation. Befehlsformat:

END_SURFACE (ohne Parameter)

- GEO_AREA -

Erzeugt einen Menüeintrag im Airport-Menü des FS5. Bezeichnet eine "Gegend", der wiederum bestimmte Flughäfen im selben Menü zugeordnet werden können. Der verwendete Name darf nicht länger als 25 Zeichen sein. GEO_AREA darf nicht mehr als einmal verwendet werden. Befehlsformat:

GEO_AREA "<Name der Gegend">

- GROUND_COLOUR -

Setzt einen Farbwert für nachfolgende Zeichenoperationen. Wird insbesondere für Straßen, Taxiways und Bodenflächen verwendet. GROUND_COLOUR ist eine einfache Operation. Befehlsformat:

GROUND_COLOUR <Wert aus [0..255]>

- ID -

Ordnet dem erzeugten Szeneriefile eine Ident-Nummer zu. Wird für Einträge in Szeneriebibliotheken verwendet. Muß in jedem Quelltext verwendet werden. Befehlsformat:
ID <Wert aus [1..255]>

- ILS -

Vereinbart an spezifizierter absoluter Koordinate ein ILS. Im Parameter "Flags" können folgende Features spezifiziert werden:

Wert	Bedeutung
1	DME
64	Glideslope
128	Sensitivity

Um mehrere Features zu erhalten, sind die jeweiligen Werte aufzuaddieren und nach "Flags" zu schreiben. Befehlsformat :

ILS <Frequenz in MHz>, <Reichweite in m>, <magnetische Abweichung in grad>,
<Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>,
"<ID, Länge <=5>", "<Name, Länge <=24>", "<Flags aus [0..255]>", <Approach Course in grad>,
<Glideslope Latitude in (G,M,S)>, <Glideslope Longitude in (G,M,S)>, <Glideslope Altitude in m>,
<Glideslope Angle in grad aus [-14.3275°... +14.3275°]>

- INCLUDE -

Fügt spezifizierte Datei in Quelltext ein. Kann zum Beispiel zum Einbinden von vorher definierten PROCs verwendet werden. Eine Datei, die durch "INCLUDE" eingebunden wird, darf selbst keine INCLUDEs enthalten. Jeder einzubindende Quelltext muß mit "END" abschließen. Befehlsformat:

INCLUDE "<Name der einzulesenden Datei>"

- INLINE -

Fügt direkt Bytefolge in den Abschnitt des Szeneriefiles ein, in dem die einfachen Operationen gehalten werden. INLINE ist eine einfache Operation. Befehlsformat:

INLINE <Zahl aus [0..255]>, <Zahl aus [0..255]>, <Zahl aus [0..255]>, ...

- INNER_MARKER -

Setzt einen "Inner"-Marker an spezifizierter Absolutkoordinate. Befehlsformat:

INNER_MARKER <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>

- JUMP -

Führt innerhalb des Abschnitts des Szeneriefiles, in dem die einfachen Operationen gehalten werden, einen Absolutsprung an spezifizierte Sprungmarke aus. JUMP ist eine einfache Operation. Befehlsformat :

JUMP "<Name der Sprungmarke>"

- JUMP_ON_VAR_NOT -

Führt innerhalb des Abschnitts des Szeneriefiles, in dem die einfachen Operationen gehalten werden, einen bedingten Sprung an die durch die Sprungmarke spezifizierte Stelle aus. Der Sprung wird dann ausgeführt, wenn die durch die Nummer spezifizierte interne Variable des FS5 nicht den angegebenen Wert aufweist. JUMP_ON_VAR_NOT ist eine einfache Operation. Befehlsformat:

JUMP_ON_VAR_NOT <Var.-Nr.>, <Wert, der über NOT mit der Variable verknüpft werden soll>,
"<Name der Sprungmarke>"

- JUMP_ON_VAR_RANGE -

Führt innerhalb des Abschnitts des Szeneriefiles, in dem die einfachen Operationen gehalten werden, einen bedingten Sprung an die durch die Sprungmarke spezifizierte Stelle aus. Der Sprung wird dann ausgeführt, wenn die durch die Nummer spezifizierte interne Variable des FS5 innerhalb des spezifizierten Wertebereichs liegt. JUMP_ON_VAR_RANGE ist eine einfache Operation. Befehlsformat:

JUMP_ON_VAR_RANGE <Var.-Nr.>, <Startwert des Wertebereichs, in dem die Variable liegen soll>,
<Endwert des Wertebereichs, in dem die Variable liegen soll>,
"<Name der Sprungmarke>"

- JUMP_ON_VECTOR -

Führt innerhalb des Abschnitts des Szeneriefiles, in dem die einfachen Operationen gehalten werden, einen bedingten Sprung an die durch die Sprungmarke spezifizierte Stelle aus. Der Sprung wird dann ausgeführt, wenn die Augenblicksposition weiter vom gerade dargestellten Gebiet in Richtung des spezifizierten Vektors entfernt ist, als durch die Parameter "Wert1" und "Wert2" festgelegt. Meist genügt es, "Wert1" auf die zu testende Distanz einzustellen und "Wert2" auf 0 zu belassen. JUMP_ON_VECTOR ist eine einfache Operation. Befehlsformat:

JUMP_ON_VECTOR <Vektor Latitude in m>, <Vektor Longitude in m>, <Vektor Altitude in m>,
<Wert 1>, <Wert 2>, "<Name der Sprungmarke>"

- LABEL -

Vereinbart eine Sprungmarke für Sprünge und Unterprogrammaufrufe. LABEL ist eine einfache Operation. Befehlsformat:

LABEL "<Name der Sprungmarke>" *taucht im BGL File nicht auf*

- LINE_COLOUR -

Setzt einen Farbwert für nachfolgende Zeichenoperationen. Wird insbesondere für Linien verwendet.

LINE_COLOUR ist eine einfache Operation. Befehlsformat:

LINE_COLOUR <Wert aus [0..255]>

- LINE_TO -

Zeichnet von zuletzt gesetztem Punkt (per MOVE_TO bzw. LINE_TO) zu neuem spezifizierten Punkt. Die Zielkoordinaten sind relativ zum vorher spezifizierten Referenzpunkt anzugeben. LINE_TO ist eine einfache Operation. Befehlsformat:

LINE_TO <rel. Latitude in m>, <rel. Longitude in m>, <rel. Altitude in m>

- LINE_TO_POINT -

Zeichnet vom zuletzt gesetzten Punkt (per MOVE_TO_POINT bzw. LINE_TO_POINT) zu neuem durch Index spezifizierten Punkt. Der Punktindex muß im selben Sichtbarkeitsbereich durch DEFINE_POINTS oder DEFINE_ROTATED_POINTS definiert worden sein. LINE_TO_POINT ist eine einfache Operation. Befehlsformat:

LINE_TO_POINT <Punktindex>

- MIDDLE_MARKER -

Setzt einen "Middle"-Marker an spezifizierter Absolutkoordinate. Befehlsformat:

MIDDLE_MARKER <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>

- MOVE_TO -

Verschiebt Zeichenzeiger auf neuen spezifizierten Punkt. Die Zielkoordinaten sind relativ zum vorher definierten Bezugspunkt anzugeben. MOVE_TO ist eine einfache Operation. Befehlsformat:

MOVE_TO <rel. Latitude in m>, <rel. Longitude in m>, <rel. Altitude in m>

- MOVE_TO_POINT -

Verschiebt Zeichenzeiger auf neuen durch Index spezifizierten Punkt und legt Linienbreite in m für nachfolgende LINE_TO_POINT-Befehle fest. Der Punktindex muß im selben Sichtbarkeitsbereich durch DEFINE_POINTS oder DEFINE_ROTATED_POINTS definiert worden sein. MOVE_TO_POINT ist eine einfache Operation.

Befehlsformat:

MOVE_TO_POINT <Punktindex>

- NDB -

Vereinbart ein ungerichtetes Funkfeuer (NDB) an spezifizierter Absolutkoordinate. Die Frequenz ist in KHz anzugeben und darf auch einen Nachkommaanteil aufweisen; dabei werden aber nur Nachkommaanteile .0 oder .5 korrekt eingelesen. Befehlsformat:

NDB <Frequenz in kHz>, <Reichweite in 2048-m-Einheiten>, <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>, "<ID, Länge<= 5>"; "<Name, Länge<= 24>"

- OUTER_MARKER -

Setzt einen "Outer"-Marker an spezifizierter Absolutkoordinate. Befehlsformat:

OUTER_MARKER <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>

- PALETTE -

Spezifiziert zu verwendende Farbpalette für nachfolgende primitive Operationen. Die entsprechende Paletten-Datei muß im Texture-Directory des FS5 stehen. Der Parameter "Dummy" ist bis auf weiteres auf "0" zu setzen. PALETTE ist eine einfache Operation. Befehlsformat:

PALETTE "<Name der Paletten-Datei, mit Extension!>", <Dummy aus [0..65535]>

- POINT -

Setzt einen blinkenden Punkt relativ zum vorher spezifizierten Referenzpunkt. Der Punkt blinkt in zwei verschiedenen Farben, die ebenfalls als Parameter anzugeben sind. Kann als Orientierungshilfe dienen. POINT ist eine einfache Operation. Befehlsformat:

POINT <rel. Latitude in m>, <rel. Longitude in m>, <rel. Altitude in m>,

<Farbe 1 aus [0..255]>, <Farbe 2 aus [0..255]>

denselbe Code wie BLINK

- POLY -

Zeichnet ein Polygon, beschrieben durch die Nummern der das Polygon begrenzenden Punkte. Die Punkte sind vorher mithilfe von DEFINE_POINTS oder DEFINE_ROTATED_POINTS im selben Sichtbarkeitsbereich zu

vereinbaren. Neben den Punkten muß ein Distanzvektor (mit Latitude-, Longitude- und Altitude-Komponente) angegeben werden, der angibt, wann das betreffende Polygon gezeichnet werden soll; dies ist insbesondere zur Unterdrückung von nicht sichtbaren Seiten eines Gebäudes erforderlich. Der Distanzvektor kann auch automatisch durch das FS5DesignerTool berechnet werden; dann sind die drei Komponenten ("Delta Latitude", "Delta Longitude" und "Delta Altitude") auf "0" zu setzen. Ferner ist ein Vektor anzugeben, der bezogen auf das Polygon nach "außen" weist; der Vektor darf nicht länger als 32767 sein. Der Vektor kann auch automatisch durch das FS5DesignerTool berechnet werden; dann sind die drei Komponenten ("Latitude Vektor" / "Longitude Vektor" / "Altitude Vektor") mit "0" anzugeben, und das Polygon ist unbedingt im Uhrzeigersinn zu definieren. Der automatisch erzeugte Vektor steht senkrecht auf dem Polygon und hat die Länge 32767. Das Polygon wird relativ zum aktuellen Bezugspunkt gezeichnet. POLY ist eine einfache Operation. Befehlsformat:

POLY <Anzahl der Punkte>, <Delta Latitude>, <Delta Longitude>, <Delta Altitude>,
 <Latitude Vektor>, <Longitude Vektor>, <Altitude Vektor>,
 <Indices der Punkte, getrennt durch Kommata>

- POLYGON -

Zeichnet ein dreidimensionales Polygon, beschrieben durch die Nummern der das Polygon begrenzenden Punkte. Die Punkte sind vorher mithilfe von DEFINE_POINTS oder DEFINE_ROTATED_POINTS im selben Sichtbarkeitsbereich zu vereinbaren. Neben den Punkten kann optional ein Distanzwert im Bereich [0..2^31] anzugeben, der dazu dient, dem FS5 mitzuteilen, welche "Seite" eines Objekts gerade sichtbar ist. Dieser Wert wird nur dann gelesen, wenn ihm ein Semikolon (anstatt eines Kommas) vorangestellt wird; andernfalls kann er weggelassen werden. Wird der Distanzwert im Befehl weggelassen oder zu "0" gesetzt, berechnet das FS5DesignerTool automatisch den korrekten Wert. Weiter ist ein Vektor anzugeben, der bezogen auf das Polygon nach "außen" weist; der Vektor darf nicht länger als 32767 sein. Der Vektor kann auch automatisch durch das FS5DesignerTool berechnet werden; dann sind die drei Komponenten mit "0" anzugeben, und das Polygon ist unbedingt im Uhrzeigersinn zu definieren. Der automatisch erzeugte Vektor steht senkrecht auf dem Polygon und hat die Länge 32767. Das Polygon wird relativ zum aktuellen Bezugspunkt gezeichnet. POLYGON ist eine einfache Operation. Befehlsformat:

POLYGON <Anzahl der Punkte>

[; <Distanzwert im Bereich [0..2^31]>], ' optionaler Parameter, kann auch entfallen!
 <Latitude Vektor>, <Longitude Vektor>, <Altitude Vektor>,
 <Indices der Punkte, getrennt durch Kommata>

- PRE_USE_BITMAP -

Muß manchmal vor dem Befehl "USE_BITMAP" verwendet werden, damit die Bitmap sichtbar wird.

PRE_USE_BITMAP ist eine einfache Operation. Befehlsformat:

PRE_USE_BITMAP (ohne Parameter)

- PROC -

Vereinbart eine Codesequenz aus beliebigen FS5DesignerTool-Operationen, die später im Programmtext (auch mehrmals) "eingebunden" werden kann. Bei Einbinden des so definierten Codes mithilfe CALLPROC wird der Code textuell in den Quelltext eingefügt; durch diese Technik wird also kein Platz im Szenieriefle gespart, sondern nur die Lesbarkeit der Quelltexte verbessert. Jedem "PROC"- Programmstück ist ein eindeutiger Name zuzuordnen, sowie die Anzahl der als Parameter zu übergebenden Zahlenwerte. Parameter lassen sich innerhalb des "PROC"-Programmstücks mit "%"+(Nr. des Parameters) angeben. Der erste Parameter ist also %1, der zweite %2, usw. Es können nur Zahlenwerte als Parameter übergeben werden, keine Strings. "PROC"- Programmstücke dürfen nicht geschachtelt werden. Innerhalb eines "PROC"-Programmstücks darf kein "CALLPROC" gemacht werden. Befehlsformat:

PROC <eindeutiger Name>-, <Anzahl der Parameter>

- RANGE -

Legt für das Szenieriefle die Extremwerte der Latitude- und Longitude-Koordinaten fest. Operationen im Quelltext, die außerhalb dieser "Range"-Werte liegen, werden vom FS5 nicht ausgeführt. Dieser Befehl muß in jedem FS5DesignerTool-Quelltext verwendet werden. Befehlsformat:

RANGE <Min. Latitude in (G,M,S)>, <Max. Latitude in (G,M,S)>,
 <Min. Longitude in (G,M,S)>, <Max. Longitude in (G,M,S)>

Defpunkt → 5.28

- REF_POINT -

Definiert einen Referenzpunkt (den Nullpunkt) sowie einen Skalierungsfaktor und -divisor für die nachfolgenden einfachen Operationen des aktuellen Sichtbarkeitsbereiches. Faktor und Divisor sind nicht bei allen einfachen Operationen wirksam. Im allgemeinen benötigt es, Faktor und Divisor auf "1" zu setzen; dann ist die Basiseinheit 1 m. Daneben kann optional ein Sprungziel angegeben werden, wohin das der FS5 springen soll, wenn er den aktuellen Sichtbarkeitsbereich nicht liest. Wird kein Sprungziel angegeben, so setzt das FS5DesignerTool das Sprungziel auf den nächsten AREA_TEST. Innerhalb von Unterprogrammen ist es dagegen notwendig, das Sprungziel auf die abschließende "RETURN"-Anweisung zu setzen. REF_POINT ist für das Funktionieren der meisten anderen einfachen Operationen notwendig. REF_POINT ist eine einfache Operation. Befehlsformat:

REF_POINT <Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>,

<Skalierungsfaktor>, <Skalierungsdivisor>

[, "<Name des Sprungziels>"]

'optionaler Parameter

*taucht in BGL hier nicht auf***- REL_AREA_TEST1 -**

Legt für einen zusammenhängenden Teil der einfachen Operationen einen zusammenhängenden Sichtbarkeitsbereich fest. Die Koordinate des Sichtbarkeitsbereichs ist aber nicht - wie beim normalen AREA_TEST - absolut anzugeben, sondern als Verschiebung, bezogen auf den zuletzt gemachten Sichtbarkeitstest. Die Verschiebung ist als Distanz in m und als Richtungswinkel in Grad anzugeben. Daneben wird die Größe des Bereichs, nach oben abgeschätzt, im Parameter "Range" beschrieben; dieser Wert ist in Einheiten von 2048 m anzugeben. Zum korrekten Abarbeiten der Szenieriefles ist bei Verwendung von einfachen Operationen ein Befehl vom Typ AREA_TEST zu benutzen, da der FS5 die Szenerie sonst gar nicht lesen kann. Befehlsformat:

REL_AREA_TEST1 <Verschiebungsdistanz in m>, <Verschiebungswinkel in grad>, <Range aus [1..255]>

- REL_AREA_TEST2 -

Legt für einen zusammenhängenden Teil der einfachen Operationen einen zusammenhängenden Sichtbarkeitsbereich fest. Die Koordinate des Sichtbarkeitsbereichs ist aber nicht -- wie beim normalen AREA_TEST -- absolut anzugeben, sondern als Verschiebung, bezogen auf den zuletzt gemachten Sichtbarkeitstest. Die Verschiebung ist in m anzugeben, aufgelöst in Latitude, Longitude, ähnlich einer relativen Koordinate. Daneben wird die Größe des Bereichs, nach oben abgeschätzt, im Parameter "Range" beschrieben; dieser Wert ist in Einheiten von 2048 m anzugeben. Zum korrekten Abarbeiten der Szenieriefles ist bei Verwendung von einfachen Operationen ein Befehl vom Typ AREA_TEST zu benutzen, da der FS5 die Szenerie sonst gar nicht lesen kann. Befehlsformat:

REL_AREA_TEST2 <rel. Latitude in m>, <rel. Longitude in m>, <Range aus [1..255]>

-REL_REF_POINT1 -

Definiert einen Referenzpunkt (den Nullpunkt) sowie einen Skalierungsfaktor und -divisor für die nachfolgenden einfachen Operationen des aktuellen Sichtbarkeitsbereiches. Faktor und Divisor sind nicht bei allen einfachen Operationen wirksam. Im allgemeinen genügt es, Faktor und Divisor auf "1" zu setzen; dann ist die Basiseinheit 1 m. Der Referenzpunkt wird aber nicht als absolute Koordinate angegeben, sondern als Verschiebung, bezogen auf den zuletzt gesetzten Referenzpunkt. Die Verschiebung ist als Distanz (in m) und Verschiebungswinkel (in Grad) anzugeben. Daneben kann optional ein Sprungziel angegeben werden, wohin das der FS5 springen soll, wenn er den aktuellen Sichtbarkeitsbereich nicht liest. Wird kein Sprungziel angegeben, so setzt das FS5DesignerTool das Sprungziel auf den nächsten AREA_TEST. Innerhalb von Unterprogrammen ist es dagegen notwendig, das Sprugziel auf die abschließende "RETURN"-Anweisung zu setzen. Das Setzen eines Referenzpunktes ist für das Funktionieren der meisten anderen einfachen Operationen notwendig.

REL_REF_POINT1 ist eine einfache Operation. Befehlsformat:

REL_REF_POINT1 <Verschiebungsdistanz in m>, <Richtung in grad>, <Altitude in m>,
 <Skalierungsfaktor>, <Skalierungsdivisor>
 [, "<Name des Sprungziels>"] 'optionaler Parameter

- REL_REF_POINT2 -

Definiert einen Referenzpunkt (den Nullpunkt) sowie einen Skalierungsfaktor und -divisor für die nachfolgenden einfachen Operationen des aktuellen Sichtbarkeitsbereiches. Faktor und Divisor sind nicht bei allen einfachen Operationen wirksam. Im allgemeinen genügt es, Faktor und Divisor auf "1" zu setzen; dann ist die Basiseinheit 1 m. Der Referenzpunkt wird aber nicht als absolute Koordinate angegeben, sondern als Verschiebung, bezogen auf den zuletzt gesetzten Referenzpunkt. Die Verschiebung ist in m anzugeben, aufgeteilt in Latitude, Longitude und Altitude, wie eine relative Koordinate. Daneben kann optional ein Sprungziel angegeben werden, wohin das der FS5 springen soll, wenn er den aktuellen Sichtbarkeitsbereich nicht liest. Wird kein Sprungziel angegeben, so setzt das FS5DesignerTool das Sprungziel auf den nächsten AREA_TEST. Innerhalb von Unterprogrammen ist es dagegen notwendig, das Sprugziel auf die abschließende "RETURN"-Anweisung zu setzen. Das Setzen eines Referenzpunktes ist für das Funktionieren der meisten anderen einfachen Operationen notwendig. REL_REF_POINT2 ist eine einfache Operation. Befehlsformat:

REL_REF_POINT2 <rel. Latitude in m>, <rel. Longitude in m>, <rel. Altitude in m>,
 <Skalierungsfaktor>, <Skalierungsdivisor>
 [, "<Name des Sprungziels>"] 'optionaler Parameter

- RETURN -

Kehrt aus einem Unterprogramm zurück innerhalb des Abschnitts des Szenieriefles, in dem die einfachen Operationen gehalten werden. Muß am Ende jedes solchen Unterprogramms stehen. RETURN ist eine einfache Operation. Befehlsformat :

RETURN (ohne Parameter)

- ROAD_DRAW1 -

Ermöglicht die bequeme Definition von Straßen des Typs 1. Ausgehend von einem Startpunkt relativ zum vorher definierten Referenzpunkt wird jedes nachfolgende Teilstück über seine Länge in m und einen Winkel definiert, der die Orientierung des betreffenden Teilstücks darstellt. ROAD_DRAW1 ist eine einfache Operation. Befehlsformat:

ROAD_DRAW1 <rel.Startlat.in m>, <rel.Startlong.in m>, <rel.Startalt.in m>,
 <Breite der Straße in m>, <Anzahl n der Teilstücke>,
 <Länge des 1.Teilstücks in m>, <Richtung des 1.Teilstücks in grad>,
 ...
 <Länge des n.Teilstücks in m>, <Richtung des n.Teilstücks in grad>

*Makro Befehl;
 wird umgesetzt
 in MOVE_TO +
 LINE_TO*

- ROAD_DRAW2 -

Ermöglicht die bequeme Definition von Straßen des Typs 2. Ausgehend von einem Startpunkt relativ zum vorher definierten Referenzpunkt wird jedes nachfolgende Teilstück über seine Länge in m und einen Winkel definiert, der die Orientierung des betreffenden Teilstücks darstellt. ROAD_DRAW2 ist eine einfache Operation.

Befehlsformat:

ROAD_DRAW2 <rel.Startlat.in m>, <rel.Startlong.in m>, <rel.Startalt.in m>,
 <Breite der Straße in m>, <Anzahl n der Teilstücke>,
 <Länge des 1.Teilstücks in m>, <Richtung des 1.Teilstücks in grad>,
 ...
 <Länge des n.Teilstücks in m>, <Richtung des n.Teilstücks in grad>

wie ROAD_DRAW1

- ROAD_LINE_TO1 -

Ermöglicht das Zeichnen von Straßen des Typs 1. Zeichnet von zuletzt gesetztem Punkt (per ROAD_MOVE_TO1 bzw. ROAD_LINE_TO1) zu neuem spezifizierten Punkt. Die Zielkoordinaten sind relativ zum vorher spezifizierten Referenzpunkt anzugeben. ROAD_LINE_TO1 ist eine einfache Operation. Befehlsformat:

ROAD_LINE_TO1 <rel.Lat.in m>, <rel.Long.in m>, <rel.Alt.in m>

- ROAD_LINE_TO2 -

Ermöglicht das Zeichnen von Straßen des Typs 2. Zeichnet von zuletzt gesetztem Punkt (per ROAD_MOVE_TO2 bzw. ROAD_LINE_TO2) zu neuem spezifizierten Punkt. Die Zielkoordinaten sind relativ zum vorher spezifizierten Referenzpunkt anzugeben. ROAD_LINE_TO2 ist eine einfache Operation. Befehlsformat:

ROAD_LINE_TO2 <rel.Lat.in m>, <rel.Long.in m>, <rel.Alt.in m>

- ROAD_MOVE_TO1 -

Verschiebt Zeichenzeiger für Straßen des Typs 1 auf neuen spezifizierten Punkt und legt Straßenbreite in m für nachfolgende ROAD_LINE_TO1-Befehle fest. Die Zielkoordinaten sind relativ zum vorher definierten Bezugspunkt anzugeben. ROAD_MOVE_TO1 ist eine einfache Operation. Befehlsformat:

ROAD_MOVE_TO1 <rel.Lat.in m>, <rel.Long.in m>, <rel.Alt.in m>, <Breite in m>

- ROAD_MOVE_TO2 -

Verschiebt Zeichenzeiger für Straßen des Typs 2 auf neuen spezifizierten Punkt und legt Straßenbreite in m für nachfolgende ROAD_LINE_TO2-Befehle fest. Die Zielkoordinaten sind relativ zum vorher definierten Bezugspunkt anzugeben. ROAD_MOVE_TO2 ist eine einfache Operation. Befehlsformat:

ROAD_MOVE_TO2 <rel.Lat.in m>, <rel.Long.in m>, <rel.Alt.in m>, <Breite in m>

- ROTATED_CALL -

Ermöglicht das Aufrufen eines durch eine Marke referenzierten Unterprogramms aus einfachen Operationen, wobei alle einfachen Operationen mit einem um für die x-, y- und z-Achsen spezifizierten Winkel ausgeführt werden. Wie jedes Unterprogramm ist auch eine Sequenz, die durch AUTO_CALL aufgerufen wird, mit "RETURN" abzuschließen. AUTO_CALL ist eine einfache Operation. Befehlsformat:

ROTATED_CALL <x-Winkel in grad>, <y-Winkel in grad>, <z-Winkel in grad>, "<Name der Marke>"

- RUNWAY -

Definiert eine Landebahn an absoluter Koordinate. Im Parameter "ID" kann eine auf der Runway dargestellte Zahl im Bereich [1..63] spezifiziert werden; soll daneben auch einer der Buchstaben "L", "R" oder "C" auftauchen, so ist dem Parameter "ID" der Wert 64 ("L"), 128 ("R") oder 192 ("C") hinzuzuaddieren. Im Parameter "Oberfläche" können 4 Typen spezifiziert werden:

Wert	Typ
0	graues Muster
1	betonartige Oberfläche, mit Punkten
2	betonartige Oberfläche, dunkel, mit Streifen
3	grasartiges Oberfläche

Im Parameter "Markers" können Markierungen auf der Landebahn spezifiziert werden; diese können alle auch durch Addition der Wertigkeiten kombiniert werden (außer "8" und "64"):

Wert	Bedeutung
1	Seitenlinien
2	Schwellenmarkierungen
4	Landemarkierungen (3 schmale Linien)
8	Abstandsmarkierungen (breite Linien)

16	Mittellinie
32	Darstellung der ID-Zahl
64	Abstandsmarkierungen (schmale Linien)

Im Parameter "Lights" kann Beleuchtung an den Seiten und der Mitte der Landebahn eingestellt werden:

Wert	Bedeutung
0	keine Beleuchtung
1	Beleuchtung an den Seiten
4	Beleuchtung in der Mitte
5	Beleuchtung an den Seiten und in der Mitte

Für beide Enden der Landebahn können Schwellenmarkierungen, Markierungsbereiche, Anflugbefeuerungen und Stroboskoplichter spezifiziert werden. Die Schwellenmarkierungen sind nur an (1) oder aus (0) zu schalten. Für jedes Ende der Landebahn kann einer von zwei möglichen Markierungsbereichen definiert werden; es darf also immer nur einer der beiden Typen auf einer Seite verwendet werden. Empfohlene Werte sind "0" (kein Markierungsbereich) und "1010" (Markierungsbereich der Länge 1010 Fuß), bei anderen Werten kann es passieren, daß der FS5 nicht mehr korrekt arbeitet. In den Parametern "Anflugbefeuerung" kann aus drei verschiedene Typen ausgewählt werden:

Wert	Typ
2	Anflugbefeuerung mittlerer Intensität (MALSR)
6	ALSF-1
7	ALSF-2

In den Parametern "Strobezahl" kann für beide Enden die Anzahl der Stroboskoplichter angegeben werden. Für beide Seiten der Landebahn können VASIs vereinbart werden. In den Parametern "VASI 1" und "VASI 2" ist festzulegen, wieviele VASI-Balken jeweils dargestellt werden sollen:

Wert	Anzahl VASI-Balken
0	0
1	2
2	3

In den Parametern "Winkel VASI 1" und "Winkel VASI 2" ist die Ausrichtung der VASIs anzugeben. RUNWAY ist eine einfache Operation. Befehlsformat:

RUNWAY <Latitude in (G,M,S), <Longitude in (G,M,S)>, <Altitude in m>, <Orientierung in grad>,
 <Länge in Fuß>, <Breite in Fuß>, <ID aus [0..255]>, <Oberfläche aus [0..3]>,
 <Markers aus [0..255]>, <Lights aus {0,1,4,5}>,
 <End 1 Schwellenmarkierungen aus {0,1}>, <End 1 Länge des Markierungsbereichs Typ 1 in Fuß>,
 <End 1 Länge des Markierungsbereichs Typ 2 in Fuß>, <End 1 Anflugbefeuerung aus {2,6,7}>,
 <End 1 Strobezahl>,
 <End 2 Schwellenmarkierungen aus {0,1}>, <End 2 Länge des Markierungsbereichs Typ 1 in Fuß>,
 <End 2 Länge des Markierungsbereichs Typ 2 in Fuß>, <End 2 Anflugbefeuerung aus {2,6,7}>,
 <End 2 Strobezahl>,
 <VASIS 1 aus {0,1,2}>, <Winkel VASIS 1 in grad>,
 <VASIS 2 aus {0,1,2}>, <Winkel VASIS 2 in grad>

- SCENERY -

Erzeugt einen Eintrag im FS5-Menü "Szenarien" unter dem Menüpunkt "Szeneriebibliothek", mit Namen und Beschreibung der Szenerie. Es sind fünf erlaubte ID-Werte aus [0..255] anzugeben, nicht benötigte ID-Plätze können auf "0" gesetzt werden. Dieser Befehl darf nur einmal verwendet werden. Befehlsformat:

SCENERY <ID1>, <ID2>, <ID3>, <ID4>, <ID5>, "<Name>", "<Beschreibung>"

→ es wird eine -VIS-Datei generiert

- SET_VAR -

Verändert eine durch ihre Nummer spezifizierte interne Variable des FS5. SET_VAR ist eine einfache Operation. Befehlsformat:

SET_VAR <Variablen-Nummer>, <gewünschter Wert>

- SHADE_COLOUR -

Setzt einen Farbwert für nachfolgende Zeichenoperationen. Wird insbesondere für solide Objekte verwendet. SHADE_COLOUR ist eine einfache Operation. Befehlsformat:

SHADE_COLOUR <Wert aus [0..255]>

- SHADED_POLYGON -

Zeichnet ein dreidimensionales Polygon mit Farbverläufen, beschrieben durch die Nummern der das Polygon begrenzenden Punkte. Zur Erzielung des Farbverlaufs ist der Befehl "SHADE_COLOUR" voranzustellen. Die Punkte sind vorher mithilfe von DEFINE_POINTS oder DEFINE_ROTATED_POINTS im selben Sichtbarkeitsbereich zu vereinbaren. Neben den Punkten kann optional ein Distanzwert im Bereich [0..2^31] anzugeben, der dazu dient, dem FS5 mitzuteilen, welche "Seite" eines Objekts gerade sichtbar ist. Dieser Wert wird nur dann gelesen, wenn ihm ein Semikolon (anstatt eines Kommas) vorangestellt wird; anderenfalls kann er weggelassen werden. Wird der Distanzwert im Befehl weggelassen oder zu "0" gesetzt, berechnet das FS5DesignerTool automatisch den korrekten Wert. Weiter ist ein Vektor anzugeben, der bezogen auf das Polygon nach "außen" weist; der Vektor darf nicht länger als 32767 sein. Der Vektor kann auch automatisch durch das FS5DesignerTool berechnet werden; dann sind die drei Komponenten mit "0" anzugeben, und das

Polygon ist unbedingt im Uhrzeigersinn zu definieren. Der automatisch erzeugte Vektor steht senkrecht auf dem Polygon und hat die Länge 32767. Das Polygon wird relativ zum aktuellen Bezugspunkt gezeichnet.

SHADED_POLYGON ist eine einfache Operation. Befehlsformat :

SHADED_POLYGON <Anzahl der Punkte>

[; <Distanzwert im Bereich [0..2^31]>], ' optionaler Parameter, kann auch entfallen!
<Latitude Vektor>, <Longitude Vektor>, <Altitude Vektor>,
<Indices der Punkte, getrennt durch Kommata>

- SHADOW_CALL -

Ruft ein Unterprogramm aus einfachen Operationen auf und zeichnet für alle darin definierten Objekte einen Schatteneffekt. Das Unterprogramm ist mit "RETURN" abzuschließen. Dieser Effekt ist nicht auf allen Grafik-konfigurationen des FS5 wirksam. SHADOW_CALL ist eine einfache Operation. Befehlsformat :

SHADOW_CALL "<Name der Marke>"

- START_SUB -

Signalisiert dem FS5, daß die nachfolgende Codesequenz aus einfachen Operationen ein Unterprogramm darstellt. START_SUB ist eine einfache Operation. Befehlsformat:

START_SUB (ohne Parameter)

- STRIPE -

Ermöglicht das Definieren von mehreren aufeinanderfolgenden Landschafts-Blöcken für einen festen Latitude-Wert innerhalb eines Intervalls von zwei Longitude-Werten. Die Bedeutungen der Parameter "Latitude", "Longitude1", "Longitude2", "Section", "Landschaftstyp", "Selektor1" bis "Selektor3" werden beim Befehl "TERRAIN" genau beschrieben. Befehlsformat:

STRIPE <Latitude in der Größe der Blöcke>,
<Longitude-Startwert in der Größe der Blöcke>, <Longitude-Endwert in der Größe der Blöcke>,
<Section aus [1..6]>,
<Altitude in m für 1. Block>, <Landschaftstyp aus [0,1,2,3] für 1.Block>,
<Selektor1 f.1. Block>, <Selektor2 f.1.Block>, <Selektor 3 f.1.Block>
....
<Altitude in m für letzten Block>, <Landschaftstyp aus [0,1,2,3] für letzten Block>,
<Selektor1 f.letzten Block>, <Selektor2 f.letzten Block>, <Selektor3 f.letzten Block>

- TAXI_DRAW -

Ermöglicht die bequeme Definition von Taxiways. Ausgehend von einem Startpunkt relativ zum vorher definierten Referenzpunkt wird jedes nachfolgende Teilstück über seine Länge in m und einen Winkel definiert, der die Orientierung des betreffenden Teilstücks darstellt. TAXI_DRAW ist eine einfache Operation. Befehlsformat:

TAXI_DRAW <rel.Startlat.in m>, <rel.Startlong.in m>, <rel.Startalt.in m>,
<Breite der Straße in m>, <Anzahl n der Teilstücke>,
<Länge des 1.Teilstücks in m>, <Richtung des 1.Teilstücks in Grad>,
...
<Länge des n.Teilstücks in m>, <Richtung des n.Teilstücks in Grad>

- TAXI_LINE_TO -

Ermöglicht das Zeichnen von Taxiways. Zeichnet von zuletzt gesetztem Punkt (per TAXI_MOVE_TO bzw. TAXI_LINE_TO) zu neuem spezifizierten Punkt. Die Zielkoordinaten sind relativ zum vorher spezifizierten Referenzpunkt anzugeben. TAXI_LINE_TO ist eine einfache Operation. Befehlsformat:

TAXI_LINE_TO <rel.Lat.in m>, <rel.Long.in m>, <rel.Alt.in m>

- TAXI_MOVE_TO -

Verschiebt Zeichenzeiger für Taxiways auf neuen spezifizierten Punkt und legt Straßenbreite in m für nachfolgende TAXI_LINE_TO-Befehle fest. Die Zielkoordinaten sind relativ zum vorher definierten Bezugspunkt anzugeben. TAXI_MOVE_TO ist eine einfache Operation. Befehlsformat:

TAXI_MOVE_TO <rel.Lat.in m>, <rel.Long.in m>, <rel.Alt.in m>, <Breite in m>

- TERRAIN -

Ermöglicht das Definieren eines einzelnen Landschafts-Blocks für einen festen Latitude-Wert und einen festen Longitude-Wert. Sowohl Latitude als auch Longitude sind hier in Abhängigkeit vom Parameter "Section" in anderen Einheiten zu interpretieren:

Section	Einheit
1	1° 24' 32"
2	42' 16"
3	21' 8"
4	10' 34"
5	5' 17"
6	2' 38.5"

Für den Block kann neben einem Höhenwert der Basislandschaftstyp festgelegt werden; dabei bedeuten

Wert	Landschaftstyp
0	Flaches Land
1	Gebirge
2	Küstenlandschaft Typ 1
3	Küstenlandschaft Typ 2

In den Selektoren 1 bis 3 kann dann weiter differenziert werden. Für den Landschaftstyp 0 existieren folgende Belegungen für Selektor 1:

Wert	Bedeutung
0	See
1	grünbraune Landschaft
2	andere grünbraune Landschaft
3	bräunliche und grüne Landschaft
4	grüne Landschaft mit Hecken
5	hellgrüne Hügellandschaft
6	braun mit wenig grün
7	braun
8	weiß
9	hellweiß
10	grünbraune Landschaft mit blauen Flecken (Seen)
11	in Parzellen aufgeteilte Landschaft mit Feldern und Wald
12	wie 11, nur dunkler
128	Feld
129	dichte Stadtlandschaft
130	schwach besiedelte Stadtlandschaft
131	Kleinstadt
132	ländliche Gegend
133	andere ländliche Gegend
134	andere ländliche Gegend
135	Weideland
136	grüne Hügellandschaft
137	braune Hügellandschaft
138	Betonlandschaft
139	andere Betonlandschaft
140	grüngrau-Gemisch
141	bewachsene Sumpflandschaft
142	Betonstadt mit Hochhäusern
143	andere Betonlandschaft

Bei anderen Belegungen von Selektor 1 für Landschaftstyp 0 verlangt der FS5 eine Texture-Datei im texture-Directory mit Namen "seed" + (Hexadezimal-Wert Selektor 1) + ".r8", also etwa "seedff.r8" bei Selektor 1 = 255 (dezimal) = FF (hexadezimal). Für den Landschaftstyp 1 (Gebirge) kann in Selektor 1 die Farbe der Berge angewählt werden:

Wert	Bedeutung
0	dunkelblau
1	dunkelgrün
2	mittelgrün
3	dunkelgrün
4	hellgrün
5	dunkelgrün
6	braun
7	braun
8	weiß
9	mittelgrün
10	schwarz
11	orange
12	orange
13	orange
14	orange
15	orange

Die Selektoren 2 und 3 sind im Normalfall auf "0" zu setzen. Befehlsformat :

TERRAIN <Latitude in der Größe der Blöcke>, <Longitude in der Größe der Blöcke>, <Section aus [1..6]>, <Altitude in m>, <Landschaftstyp aus {0,1,2,3}>, <Selektor1>, <Selektor2>, <Selektor3>

- TEX_POLYGON -

Zeichnet ein dreidimensionales Polygon mit Texturen, beschrieben durch die Nummern der das Polygon begrenzenden Punkte. Zur Anwahl der Textur ist ein Befehl "BITMAP" voranzustellen. Die Punkte sind vorher mithilfe von DEFINE_POINTS oder DEFINE_ROTATED_POINTS im selben Sichtbarkeitsbereich zu vereinbaren. Neben den Punkten kann optional ein Distanzwert im Bereich [0..2^31] anzugeben, der dazu dient, dem FS5 mitzuteilen, welche "Seite" eines Objekts gerade sichtbar ist. Dieser Wert wird nur dann gelesen, wenn ihm ein Semikolon (anstatt eines Kommas) vorangestellt wird; anderenfalls kann er weggelassen werden. Wird

der Distanzwert im Befehl weggelassen oder zu "0" gesetzt, berechnet das FS5DesignerTool automatisch den korrekten Wert. Weiter ist ein Vektor anzugeben, der bezogen auf das Polygon nach "außen" weist; der Vektor darf nicht länger als 32767 sein. Der Vektor kann auch automatisch durch das FS5DesignerTool berechnet werden; dann sind die drei Komponenten mit "0" anzugeben, und das Polygon ist unbedingt im Uhrzeigersinn zu definieren. Der automatisch erzeugte Vektor steht senkrecht auf dem Polygon und hat die Länge 32767. Das Polygon wird relativ zum aktuellen Bezugspunkt gezeichnet. SHADED_POLYGON ist eine einfache Operation. Befehlsformat:

```
TEX_POLYGON <Anzahl der Punkte>
[ ; <Distanzwert im Bereich [0..2^31]> ] , ' optionaler Parameter, kann auch entfallen!
<Latitude Vektor>, <Longitude Vektor>, <Altitude Vektor>,
<Indices der Punkte, getrennt durch Kommata>
```

- USE_BITMAP -

Zeichnet ein mit einer Bitmap gefülltes Polygon, wobei die Bitmap beliebig gestreckt und gestaucht werden kann. Die Bitmap muß vorher durch einen BITMAP-Befehl referenziert werden. Ebenso müssen die über ihre Punkt-indices verwendeten Begrenzungspunkte des Polygons vorher durch den Befehl "DEFINE_POINTS" oder "DEFINE_ROTATED_POINTS" im selben Sichtbarkeitsbereich vereinbart worden sein. Das Polygon wird nur dargestellt, wenn der aktuelle Blickpunkt höchstens um den anzugebenden Distanzwert vom Polygon entfernt ist, in Richtung des ebenfalls anzugebenden Vektors. Jedem Punkt ist ein Punkt der zweidimensionalen Bitmap-Matrix (256 * 256) zuzuordnen. USE_BITMAP ist eine einfache Operation. Befehlsformat:

```
USE_BITMAP <Anzahl der Punkte des eingrenzenden Polygons>,
<Latitude Vektor>, <Longitude Vektor>, <Altitude Vektor>,
<Distanzwert in m aus [0..2^31]>,
<Index des ersten Punktes>, <Bitmap-x des ersten Punkt 1>, <Bitmap-y des ersten Punktes>,
...
<Index des letzten Punktes>, <Bitmap-x des letzten Punktes>, <Bitmap-y des letzten Punktes>
```

- VOR -

Vereinbart an spezifizierter absoluter Koordinate ein Funkfeuer (VOR). Im Parameter "DME" kann spezifiziert werden, ob das VOR zusätzlich ein DME aufweisen soll; in diesem Fall ist als Wert "1" zu wählen, sonst 0.

Befehlsformat:

```
VOR <Frequenz in MHz>, <Reichweite in m>, <magnetische Abweichung in grad>,
<Latitude in (G,M,S)>, <Longitude in (G,M,S)>, <Altitude in m>,
"<ID, Länge <=5>","<Name, Länge <=24>","<DME aus {0,1}>"
```

B. Beispielprogramme

Im Lieferumfang des FS5DesignerTools befinden sich Beispielprogramme, anhand derer die Möglichkeiten des FS5DesignerTools demonstriert werden sollen. Für alle Dateien (außer "eurnav.fst") gilt, daß sie in der "Aerosoft Welt" existieren, und daß die zu demonstrierenden Elemente an Position (N51,36,55) und (E8,37,2) liegen.

- eurnav.fst -

Szenerie mit 1600 NDBs, ILS und VORs.

- kap??_?.fst -

Sämtliche Programmbeispiele dieses Handbuchs. Das 2. Programmbeispiel aus Kapitel 4.5 etwa heißt "kap45_2.fst".

- colours.fst -

Stellt die Palette aller 256 Farben dar in Form eines Streifens von gefärbten Polygonen. Mit fortschreitender Tageszeit werden die meisten Farben angepaßt. Farben, die zu Tages- und Nachtzeit identisch sind, können für Markierungsleuchten etc. verwendet werden.

C. Umrechnung absoluter Koordinaten in Blockkoordinaten

Die Latitude-Anteile der Blockkoordinaten verhalten sich nichtlinear zu der entsprechenden Darstellung in Absolutkoordinaten, d.h. es gibt keine einfache Formel zum Ermitteln einer Blockkoordinate aus einer Absolutkoordinate.

Wir haben hier für Blockkoordinaten des Typs 1 (niedrigste Auflösung = größte Blöcke) den Zusammenhang von Latitude-Komponente einer Blockkoordinate zur Latitude-Komponente einer absoluten Koordinate aufgelistet.

Aus Platzgründen müssen wir auf entsprechende Tabellen für die Typen 2 bis 6 verzichten. Die Tabelle für den Typ 1 kann aber leicht dazu benutzt werden, entsprechende Werte für die nachfolgenden Typen zu berechnen. Wir verweisen auf das Kapitel über Blockkoordinaten.

Alle angegebenen Koordinatenwerte wurden durch Experimente ermittelt und somit mit Fehlern behaftet.

→ Ein Block soll vermutlich immer eher quadratische Form haben

Blockwert Typ 1	Start Koordinatenbereich Lat.	Blockwert Typ 1	Start Koordinatenbereich Lat.
0	0° 0' 0"	34	43° 16' 48"
1	1° 24' 30"	35	44° 18' 16"
2	2° 49' 0"	36	45° 18' 47"
3	4° 13' 26"	37	46° 18' 19"
4	5° 37' 40"	38	47° 16' 50"
5	7° 1' 50"	39	48° 14' 8"
6	8° 25' 40"	40	49° 10' 10"
7	9° 49' 17"	41	50° 5' 35"
8	11° 12' 30"	42	50° 59' 40"
9	12° 35' 31"	43	51° 52' 59"
10	13° 58' 1"	44	52° 45' 26"
11	15° 20' 0"	45	53° 36' 14"
12	16° 41' 33"	46	54° 26' 33"
13	18° 2' 25"	47	55° 15' 30"
14	19° 22' 49"	48	56° 3' 55"
15	20° 42' 37"	49	56° 50' 56"
16	22° 1' 36"	50	57° 37' 13"
17	23° 19' 58"	51	58° 22' 25"
18	24° 37' 36"	52	59° 6' 49"
19	25° 54' 26"	53	59° 50' 1"
20	27° 10' 25"	54	60° 32' 44"
21	28° 25' 37"	55	61° 13' 57"
22	29° 41' 1"	56	61° 54' 42"
23	30° 53' 23"	57	62° 34' 33"
24	32° 5' 55"	58	63° 13' 30"
25	33° 17' 19"	59	63° 51' 34"
26	34° 28' 9"	60	64° 28' 46"
27	35° 37' 49"	61	65° 5' 4"
28	36° 46' 8"	62	65° 40' 48"
29	37° 53' 55"	63	66° 15' 31"
30	39° 0' 30"	64	66° 49' 34"
31	40° 6' 2"	65	67° 22' 51"
32	41° 10' 42"	66	67° 55' 14" (reicht bis
33	42° 14' 32"		68° 26' 53")

Die Longitude-Anteile von Blockkoordinaten lassen sich dagegen einfach aus den Longitude-Anteilen der Absolutkoordinaten berechnen. Zwischen beiden Darstellungen besteht ein linearer Zusammenhang.

Eine Longitude-Einheit des Typs 1 entspricht 5062.5", das sind 1°24'22.5".

Eine Longitude-Einheit des Typs 2 entspricht 2531.25", das sind 42'11.25".

Eine Longitude-Einheit des Typs 3 entspricht 1265.625", das sind 21'5.625".

Eine Longitude-Einheit des Typs 4 entspricht 632.8125", das sind 10'32.8125".

Eine Longitude-Einheit des Typs 5 entspricht 316.40625", das sind 5'16.40625".

Eine Longitude-Einheit des Typs 6 entspricht 158.203125", das sind 2'38.203125".

D. Index

In diesem letzten Abschnitt haben wir eine Liste von Problemstellungen für Sie zusammengestellt. Zu jeder Frage haben wir die Seitenzahl geschrieben, ab der Sie die jeweils zugehörige Lösung finden können. Weitere Fragen an uns sind willkommen.

ATIS	Wie definiere ich ein ATIS?, 32
Ausgabezeile	Was bedeutet die Ausgabe während der Übersetzung der Quelltexte?, 5
Berge	Wie benutze ich die FS5-eigenen Gebirge?, 29
Berge	Wie zeichne ich einen einfachen Berg aus Polygonen?, 21
Bibliothek	Wie erzeuge ich meine eigene Prozedurbibliothek?, 34
Bitmap	Wie benutze ich Texturen und Bitmaps?, 18
Bitmap	Wie kann ich selbstdefinierte Bodentexturen verwenden?, 28
Bitmap	Wie zeichne ich aufrechte Polygone, die mit Texturen gefüllt sind?, 19
Bodenpolygon	Wie zeichne ich ein Bodenpolygon?, 15
Dateien	Wie lagere ich große Programmstücke in externe Dateien aus?, 35
DME	Wie definiere ich ein DME?, 10
Drehung	Wie lassen sich ganze Objekte drehen?, 22
Editor	Wie erstelle ich meine Quelltexte?, 5
f.bat	Wie benutze ich das mitgelieferte Hilfsprogramm 'f.bat'?, 9
Farbe	Wie erzeuge ich Farbverläufe?, 18
Farbe	Wie färbe ich ein Szenerieelement?, 13
Flughafen	Wie definiere ich einen Eintrag im Airport-Menü?, 30
Flughöhe	Warum verändern Straßen ab einer bestimmten Flughöhe ihr Aussehen?, 14
Flughöhe	Wie kann ich meine aktuelle Flughöhe abfragen?, 24
Gebäude	Wie erzeuge ich die Gebäude aus der Chicago-Szenerie selbst?, 24
Gebäude	Wie zeichne ich ein Gebäude aus Polygonen?, 16
Himmelsrichtung	Welche Himmelsrichtung entspricht welchem Winkel?, 14
ID-Wert	Was ist eine Ident-Nummer?, 6
ID-Wert	Wie benutze ich IDs?, 36
ILS	Wie definiere ich ein ILS?, 10
Installation	Wie installiere ich das FS5DesignerTools auf Festplatte?, 5
Kommentare	Wie kann ich Kommentare in meine Quelltexte einfügen?, 8
Koordinate	Wie erhalte ich aus einer Absolutkoordinate eine Blockkoordinate?, 27
Koordinate, absolute	Wie definiere ich eine absolute Koordinate?, 6
Koordinate, Block	Wie definiere ich eine Blockkoordinate?, 7
Koordinate, relative	Wie definiere ich eine relative Koordinate?, 7
Label	Was ist ein Label?, 21
Landebahn	Wie erzeuge ich eine Landebahn?, 30
Linien	Wie erzeuge ich einfache Linien?, 15
Linien, gepunktet	Wie zeichne ich eine gepunktete Linie?, 16
Marker	Wie definiere ich einen Inner-Marker?, 11
Marker	Wie definiere ich einen Middle-Marker?, 11
Marker	Wie definiere ich einen Outer-Marker?, 11
Minimalprogramm	Was muß in jedem FS5DesignerTool-Programm stehen?, 6
NDB	Wie definiere ich ein NDB?, 10
Numerische Ausdrücke	Welche numerischen Ausdrücke dürfen im FS5DesignerTool verwendet werden?, 7
Positionieren	Wie positioniere ich mein Flugzeug im FS5 in meine Szenerie?, 6
Punkt	Wie erzeuge ich einen blinkenden Punkt?, 24
Punktliste	Wie definiere ich eine Punktliste?, 15
Punktliste	Wie definiere ich gedrehte Punkte?, 17
RANGE	Wie benutze ich den RANGE-Befehl?, 28
Referenzpunkt	Was ist ein Referenzpunkt?, 12
Schatten	Wie erzeuge ich Schatten?, 23
Sichtbarkeitsbereich	Was ist ein Sichtbarkeitsbereich?, 12
Speicher	Wieviel Speicher steht dem FS5DesignerTool zur Verfügung?, 37
Straße	Wie lassen sich schnell Straßen und Taxiways vereinbaren?, 14
Straße	Wie zeichne ich eine Straße?, 13
Strings	Wie verwende ich Zeichenketten?, 8
Szenerie laden	Wie kann ich mir die erzeugten Szenerien auf dem FS5 ansehen?, 5
Tageszeit	Wie Sorge ich dafür, daß Objekte nachts anders dargestellt werden als tagsüber?, 23
Tageszeit	Welche Farben ändern sich während der Nachtstunden?, 13
Tanken	Wie betanke ich mein Flugzeug?, 26
Taxiways	Wie definiere ich Taxiways?, 14

Überdeckung	Was tue ich, wenn zwei oder mehr Objekte sich gegenseitig überdecken?, 22
Unterprogramm	Was kann ich tun, wenn ein Unterprogramm nicht ausgeführt wird?, 21
Vektor	Was bedeutet der 'senkrechte Vektor' bei Polygonen?, 17
Version	Mit welchen Versionen des MS Flight Simulator arbeitet das FS5DesignerTool?, 5
Verzweigung	Was ist eine bedingte Verzweigung?, 23
VOR	Wie definiere ich ein VOR?, 9
Vorrangregeln	Welche Vorrangregeln gelten bei numerischen Ausdrücken?, 7
Welt	Wie definiere ich meine eigene 'Welt'?, 36
Welt	Wie komme ich in die 'Aerosoft Welt'?, 9
Welt	Welche ID-Werte werden für Szenarien in der FS5-Welt verwendet?, 6
Zahlen	Wie groß dürfen Zahlen im FS5DesignerTool sein?, 8
Zeichenketten	Wie verwende ich Strings?, 8

RANGE-Default 5.28
 Section 5.27

Post Scriptum

...schön, daß Sie bis hierhin "durchgehalten" haben. Während Sie dieses Handbuch durchgearbeitet haben, haben wir von Aerosoft bereits neue Ideen verwirklicht, haben das FS5DesignerTool um neue Befehle erweitert und neue Lösungen für offene Probleme gefunden. Daneben sind auch Sie aufgerufen, uns von Ihren Erfahrungen mit dem FS5DesignerTool zu berichten - vielleicht auch in Form von Verbesserungsvorschlägen -, denn nur so können wir Ihnen das Produkt liefern, das Sie sich wünschen.

Wenn Sie mögen, können Sie Ihre eigenen, mit dem FS5DesignerTool erzeugtem Szenarien mit anderen FS5-Freunden austauschen, indem Sie an unserer firmeneigenen Mailbox teilnehmen, die noch Anfang 1995 "ans Netz" gehen soll. Näheres dazu erfahren Sie regelmäßig in unserem "aktuell" oder direkt von Aerosoft. <

aerosoft GmbH
 Luftfahrt-Datentechnik
 Flughafen Paderborn/Lippstadt
 D-33142 Büren-Ahden

fon 02955 7119 + 7 74 41
 fax 02955 7118

